ISTITUTO ITALIANO
DI TECNOLOGIA
PATTERN ANALYSIS
AND COMPUTER VISION

DITEN
Dipartimento di Ingegneria Navale, Elettrica,
Elettronica e delle Telecomunicazioni (DITEN)
Università degli Studi di Genova

Learning by Correlation for Computer Vision Applications: from Kernel Methods to Deep Learning

Author:
Jacopo Cavazza

Supervisor:
Prof. Vittorio Murino

A thesis submitted in fulfillment of the requirements for the degree of Doctor of Philosophy

Doctoral School in "*Scienze e Tecnologie per l'Ingegneria Elettronica e delle Telecomunicazioni*"

Doctoral Course in "*Visione Computazionale, Riconoscimento e Apprendimento Automatico*" – *XXX ciclo*

March 28, 2018

"*Much learning does not teach understanding.*"

Heraclitus

Abstract

Learning to spot analogies and differences within/across visual categories is an arguably powerful approach in machine learning and pattern recognition which is directly inspired by human cognition. In this thesis, we investigate a variety of approaches which are primarily driven by correlation and tackle several computer vision applications.

First, we build on top covariance-based paradigms to capture mutual, spatio-temporal relationships conveyed by the raw data in the case of 3D action recognition from skeletal data. In particular, we propose a generalization of covariance representation to capture (more discriminant) non-linear correlations. Subsequently, we build approximated kernel machines and we learn from data how to re-weight the covariance descriptor, as to enhance its discriminative capability. In this manner, we deploy a compact model which is scalable and performs favorably with respect to state-of-the-art approaches.

Second, we model correlations across multi-modal representations of the same data in order to spot which data annotations are outliers and thus misleading for supervised learning approaches. We formalize this problem in the case of a multi-view, manifold regularized regression framework where the Huber loss is a proxy for robustness. Outliers' removal stage is embedded within a refinement scheme in which exact optimization is guaranteed with a closed-form. A broad experimental analysis certifies the effectiveness of the approach in classical regression benchmarks, learning binary classifiers from noisy labels and crowd counting problems for surveillance.

Third, we take advantage of correlation among visual domains for the sake of unsupervised domain adaptation. We propose a novel alignment technique which, unlike currently available Euclidean approaches, act on the Riemannian manifold by the estimation of geodesics. In addition to a superior performance against state-of-the-art deep architectures for domain adaptation applied to image classification tasks, the superiority of our approach is certified by a novel and unsupervised fine-tuning strategies for free hyper-parameters which is based on entropy minimization.

Finally, we also investigate when the notion of correlation has a negative impact on the learning, precisely, when over-redundancies affect data representations, ultimately yielding to overfitting. To tackle this problem, we study a popular technique, called Dropout, that is ubiquitously applied in deep learning, despite its theoretical behavior as a regularizer remains elusive. In the case of matrix factorization problems, we establish a principled connection between Principal Component Analysis and Dropout when the latter is adaptively applied with respect to desired size of the factorization. Our theoretical findings inspire a novel algorithmic variation which is able to improve upon standard dropout training for deep convolutional neural networks when applied to image classification tasks.

By exploring several facets of correlation, our thesis shows that "learning by correlation" is a versatile tool in machine learning and pattern recognition which 1) can be combined with either hand crafted (kernel) methods or deep learning approaches and 2) favorably copes with a plethora of computer vision applications.

Acknowledgements

I would like to express my sincere gratitude to my advisor Prof. Vittorio Murino for the continuous support of my Ph.D study and related research, for his patience, motivation and extremely positive attitude.

I am also thankful towards my PhD thesis reviewers Prof. Berretti and Prof. Sclaroff who throughly read my manuscript providing very useful comments.

I am sincerely grateful to all my fellow colleagues that made working hours quite enjoyable.

A special mention must be given to Pietro for his great help and for being always capable of providing valuable discussions.

Andrea must be obviously cheered too: it was my pleasure to work together with him on a variety of stimulating problems and challenges.

Last but not least, my mother and my girlfriend helped me a lot during this period of my life with their smile and the priceless support that they gave me.

Contents

List of Figures

List of Tables

To my two beautiful women:
my mother Clara
and my girlfriend Giada

Chapter 1

Introduction

Visual recognition is surely one of the most studied problems in computer vision, machine learning and pattern recognition. It can be defined as the problem of providing a provably good surrogate for the human eyes to artificially intelligent systems. In order to "make machines able to see and understand", novel algorithms and framework are usually developed in parallel with respect to the design of novel data encodings and feature representations play a crucial role, and within the previous years, the community has explored two mainstream approaches.

First, by manually engineering the data description, one allows to proficiently encode prior information or other general expertise which are postulated to be useful. Following this trend, a number of hand-crafted descriptors have been proposed and widely used for different applications: Scale Invariant Feature Transform (SIFT) [Low04], Histograms of Oriented Gradients (HOG) [DT05], Local Binary Patterns (LBP) [Aho+06] or Local Intensity Order Pattern (LIOP) [Wan+11], to name a few.

As a second paradigm, after its breakthrough in image classification [Kri+12a], deep learning has been playing a prominent role in computer vision [Sch15]. Indeed, thanks to the deployment of massive annotated datasets and the widespread of GPU-accelerated computations, deep hierarchical feature representations can be now directly learnt in a bottom-up approach, ultimately using the data to guide the extraction of the patterns which should exploited for recognizing the data itself. Despite such fact may seems appealing, a few issues arise from the fact that carrying out optimization for such deep hierarchies is a hardly non-convex problem [Goo+16].

With this respect, not only the best feature representation must be envisaged, but on parallel, a model which is capable of exploiting the descriptiveness of the feature must be deployed. In fact, extremely elaborated features require a relatively simplified model if compared with the one which is fed with much more elementary representations. In any case, an arguably powerful approach

to deploy a model on top of feature representations is to take advantage of mutual relationships. The latter task can be done either between different values of the same feature representations or, even, different feature representations applied to the same data concurrently. Indeed, by just inspecting the intrinsic values of a data encoding, a recognition system may be fooled by several sources of noise which may change those values in *absolute* terms. As a remedy to gain in robustness, one may model the *relative* correspondence between those values, ultimately measuring the amount of correlation that they exhibit.

Inspired by these observations, in this thesis, we aim at presenting a collection of different methods which all leverage on the notion of correlation under different interpretations. In fact, we deal with several machine learning models (kernel methods [Cav+16; Cav+17a], matrix factorization [Cav+17b] and deep learning [Mor+17; Cav+17c]). While doing so, we carry out patter analysis for diverse computer vision applications (action recognition [Zun+17b], image classification [Mor+17], intention prediction [Zun+17a; Zun+17c] and crowd counting [CM15; CM16]).

In either statistics or machine learning, the notion of correlation has been quantified in many different ways. Surely, the most straightforward one is to measure to what extent a certain data distribution can be fitted through a linear model. The most known descriptor following this line is the *covariance representation* [Tuz+06a; Hus+13; Min+14b; Wan+15b; Cav+16; Min+16b; Cav+17c]. In fact, in the case of two random scalar values, the covariance representation is a measure of their joint variability and moreover, its sign indicates either direct or inverse relationship, meaning that when one variable is multiplied by a scalar factor of $\alpha \neq 0$, the other one is multiplied by α or $1/\alpha$, respectively [Ric88]. Since the magnitude of the covariance is generally unbounded, its $[-1, 1]$ normalized version is called the McPearson' correlation coefficient (ρ). The latter is widely used to quantify the presence of a linear ($\rho \approx 1$) or anti/linear ($\rho \approx -1$) dependence. Nevertheless, since linear dependences may be not sufficient to capture all the patterns which are useful for the recognition stage, several alternatives are available: explicit non-linear embeddings [LV07; Vid+16a], kernel representation [Min+14b; Wan+15b; Zho+17; Cav+16] and information theory tools, such as entropy or mutual information [VW97; SB+13]).

In addition to model (linear) dependencies *within the same* data representations, one may also look for dependencies *across different* encodings which are applied on parallel to the data. Along this line, many works [Min+13; Min+14a; CM15; Min+16a; CM16] have been proposed and can be categorized with the term *multi-view learning* [Sun13]. This means that we try to accommodate our recognition model by training on different alternative representation (hereby called *views*[1]) which are simultaneously applied to the same data for encoding. Therefore, assuming that all the multiple views are compatible and each of them provide some cues for the recognition purpose, the task is derive

[1]Let us clarify that, despite the term "view" may suggest the problem of dealing with multiple cameras that acquire the scene we want to recognize from different perspectives, this is not the case. Precisely, by views, we mean different multi-modal encodings applied in tandem to represent the same raw data. For instance, we are given an image and we decide to simultaneously encode it with HOG and SIFT descriptors, afterwards combining the two with a technique which is not a bare concatenation of the two.

a paradigm where we perform a principled cumulative training which benefits from each representation by also imposing some regularity as to ensure a coherent fusion between views [CM15; CM16].

Furthermore, we can push forward the generality of the visual entities between which correlation is computed: after capturing correlations within different components of the same representations and correlations among different multi-modal representations, we now try to correlate across visual domains. In fact, since data labeling is onerous or even impossible in some cases, we assume that one domain - called *source* - is fully annotated. It is then desirable to train a model on it and then transfer it on a *target* domain even when the latter is not annotated at all. In such case, re-training from scratch on the target domain is not viable and, in addition, another difficulty arises: the so-called *domain shift* [TE11]. It refers to visual ambiguities which make the same visual category extremely different when switching from one domain to another. In the literature, the previous problem is usually referred as *unsupervised domain adaptation*: since the target is not labeled, adaptation must be done at the feature level and, eventually, one may frame such problem as semi-supervised learning where labeled and unlabeled data come from source and target domain, respectively.

Among the ways of carrying out adaptation, correlation alignment [Sun+16; SS16; MM17; Mor+18] propose to model the source and target distributions with two separated second order statistics and, afterwards, to align the two so that a classifier trained on the source can be transferred to the target, without experiencing a drop in performance. Deploying scalable and effective techniques to achieve such type of alignment is of utmost importance. Following the current mainstream implementations [SS16; MM17] the problem is casted by adding the classification loss on the source with a regularizer which penalizes the discrepancy between source and target second-order statistics. Usually, such regularization is weighted by means of a Lagrangian multiplier which needs to be cross validated in order to infer the optimal balance between loss and regularizer. Note that, due to the domain shift, usual labeled cross-validating approaches on the source may be not representative on the performance on the target. At the same time, since no labels are given on the target, direct cross-validation on it is actually impossible. However, provided that those problems are faced through the right approach, experimental evidences show that aligning correlations is a scalable and efficient manner to carry out adaptation which achieves state-of-the-art results in digits classification and in object categorization across modalities - *e.g.*, from RGB to depth.

Within all the previous problems, the notion of correlation has a positive interpretation, indeed relating to a source of information which helps in guiding the learning towards accomplishing the recognition task. However, there exist cases when correlation actually damages the learning stage. This can happen when the data are so excessively correlated that, as a result, redundancy acts as noise which hides the recognition cues we are interested in. Eventually, an overwhelming redundancy may occur when the input data representation is so disproportionate with respect to the number of examples that optimization may fail and, even if it succeeds, the generalization capabilities are quite limited due to overfitting [Bis06].

To cope with this issue, many methods have been envisaged to get rid of this problem of over-redundancy in data representation. For instance, optimization is carried out to derive a compact embedding space where the data can be projected as to ensure that the transformed components are all independent from each others. Examples of this types are Principal Component Analysis (PCA) and its variants [Vid+16a; Kes+16; SM09] promotes for statistical independence, whereas other notions of decorrelations exploit, for instance, tools from information theory [HO00].

A recently introduced technique named *dropout* [Hin+12; Sri+14], applies the same idea of decorrelating data representation to the problem of training neural networks. Indeed, since hierarchical encodings are learnt in a data-driven fashion, the problem of counteracting redundancy is indeed relevant to recover from overfitting and gain in generalization capabilities [Hin+12; Sri+14; Mor+17]. Due to the fact that a neural network is composed by a set of multiple units (arranged either in parallel or serial configurations), dropout suppresses some of those, according to a Bernoulli scheme which randomly inhibits the unit to be activated and therefore giving its contribution in adapting the network's weights during optimization (which is usually done by batch gradient descent). Therefore, dropout acts as a model ensemble, meaning that from the original network architecture, a family of multiple subnetworks is subsampled from the original one. Each of these sub-networks share the weights with the original one and optimization is carried out in such a way that only a marginal group of neurons is responsible of each update of the weights, the group of selected neurons changing at each update. Actually, this technique has been showed to act as an implicit regularizer and many recent works have tried to explain it, as to provide a better theoretical understanding of dropout [Wag+13; HL15; BS13; BS14; Wag+14; GG16; Cav+17b].

1.1 Contributions & Thesis Outline

In this thesis, we investigate several correlation-based techniques which are traverse the literature across kernel methods, factorization models and deep learning. This allows us to handle several computer vision problems: human action recognition, crowd analysis for surveillance, intention prediction and object recognition. Precisely, in this Section, we will detail the thesis' contributions.

1. It can be easily argued that capturing data correlation in terms of linear relationships only may be just suboptimal. In fact, data may exhibit some other interesting classes of correlations which are not actually appreciable with a classical covariance representation.

   To this aim, in [Cav+16], we present a rigorous and principled mathematical pipeline to recover the kernel trick for computing the covariance matrix, enhancing it to model more complex, non-linear relationships conveyed by the raw data. A solid theoretical analysis certifies that we are able to effectively compute this novel representation through a closed-form solution, ultimately devising a new descriptor which generalize the classical covariance operator which can be embedded in our formalism as a particular case. In the experiments, we validate the proposed framework

against many previous approaches in the literature, favorably scoring in terms of (improvements over previous) state-of-the-art performance.

2. Despite covariance representation is broadly used and reliable as a tool, scalability issues arise when used in tandem with max margin kernel machines. In fact, in general, the kernel function has to be evaluated for all pairs of instances inducing a Gram matrix whose complexity is quadratic in the number of samples. In this thesis we reduce such complexity to be linear by proposing a novel and explicit feature map to approximate the kernel function [Cav+17a]. This allows to train a linear classifier with an explicit feature encoding, which implicitly implements a Log-Euclidean machine in a scalable fashion. Not only we prove that the proposed approximation is unbiased, but also we work out an explicit strong bound for its variance, attesting a theoretical superiority of our approach with respect to existing ones.

   Motivated by the recent success of kernel methods based on covariance representation for the problem of action recognition from skeletal data, we carried out a broad experimental validation showing that our representation provides a compact and scalable pipeline which outperforms state-of-the-art Fourier [RR07; WM13] and Taylor-based [KK12] approximations schemes on a number of publicly available benchmark datasets for 3D action recognition.

3. Within the panorama of human action recognition, the current state-of-the-art is contended between two different paradigms: kernel-based methods and feature learning with deep (recurrent) neural networks. Both approaches show strong performances, yet exhibiting heavy, but complementary, drawbacks. Motivated by this fact, we aim at combining together the best from the two paradigms, by proposing an approach where a shallow network is fed with a covariance representation [Cav+17c]. Since the latter is provably able to capture the action kinematics, our approach directly learns from data which are the variables whose correlation best encodes action discriminants.

   Through a solid experimental analysis, we corroborate our assumption that, as long as the dynamics is effectively modeled, there is no need for the classification network to be deep nor recurrent in order to score favorably.

4. Having certified the benefits of correlating different components of the same data representation, we endow the perspective of capturing the correlation between different representations (in short, *views*) applied to the same data instance at the same time. Despite well established settings have been proposed to accommodate for that [BM98; NG00; Mus+02a; Mus+02b; Mus+06; Yu+07; Yu+11; WZ10; Sin+05; BS04; Kum+10; Kum+11; Abn02; Bal+04; WZ07], none has investigated the issue of modeling such correlation in a robust manner, as to avoid a negative impact on learning caused by noisy correlations. To this aim, we propose to leverage on the Huber loss [Hub64] which has been established as a proxy for robustness against outliers [MM00; AZ05; LLZ11; Kha+13]. However, none of this approaches [Hub64; MM00; AZ05; LLZ11; Kha+13] have achieved a

closed-form solution for optimization and always approximation needs to be performed.

Differently, in [CM15; CM16] we propose a novel closed-form solution which is broadly applicable to a general class of regularized machine learning problems. This reflects into an efficient algorithm with the advantage that we also learn from the data a hyper-parameter (related to the Huber loss), which was heuristically fixed in prior works [MM00; AZ05; LLZ11; Kha+13]. An extensive experimental evaluation is performed on statistical regression tasks, learning from noisy labels problems and crowd counting applications.

5. We exploit the notion of correlation alignment [Sun+16] to learn across datasets in the unsupervised domain adaptation problem where a model trained on a source domain needs to be transferred on a target domain. To solve this issue, we seek for the best alignment between second order statistics across the two domains. Although this approach was already proposed in the literature [Fer+13; Sun+16; SS16], we posit that the current existing implementations for it are either non-scalable (due to matrix inversion operations) or not principled since they do not actually consider the inner geometrical structure of second order statistics which are symmetric and positive definite operators (SPD). Thus, as opposed to classical Euclidean alignment, we propose to pursue a Riemannian way of carrying out alignment along geodesics so that the inner curvature of the SPD manifold is taken into account. The superiority of the proposed alignment is demonstrated through an broad experimental validating analysis.

   When aligning second order statistics across datasets, in [Mor+18], we demonstrate that, at the optimum, correlation alignment induces target entropy minimization, the converse being not true. Induced by this theoretical framework we refine the previously proposed Riemannian alignment with a novel framework, called minimal entropy correlation alignment, which can be efficiently implemented as an additive regularizer compatible with any generic classification framework. In order to balance the importance of such regularizer, we can leverage on our theoretical findings in order to obtain a fully unsupervised entropy-based criterion which, differently from usual cross-validating strategies for hyper-parameters' tuning, requires no labeled validation set.

6. In the modern paradigm of learning how to represent the data from the data itself, excessive correlations between neurons have been shown to negatively impact on generalization and leads to overfitting issues. As a remedy, dropout has been proposed [Hin+12; Sri+14]: the main idea is to randomly suppress some units in the network according to the realization r of a binary Bernoulli random variable ($r = 1$ implies the unit's kept, the latter being suppressed if $r = 0$). Dropout can be embedded within classical back-propagation training: before one gradient update, Bernoulli variables are samples and the units in the network are suppressed accordingly - thus, weights are updated only for the remaining units. For the next backward pass, Bernoulli are re-sampled again, remaining weights are updated and the procedure iterates. In testing, there is no units' suppression, inference is done on the full model by re-scaling

the weights by the expected value of the Bernoulli variable - the so called retain probability: by considering this step as a practical surrogate for averaging across all possible simplified networks, dropout can be therefore interpreted as a sort of model ensemble.

This thesis presents a theoretical analysis of dropout for matrix factorization, where Bernoulli random variables are used to drop a factor, thereby attempting to control the size of the factorization. While recent work has demonstrated the empirical effectiveness of dropout for matrix factorization, a theoretical understanding of the regularization properties of dropout in this context remains elusive. We demonstrate the equivalence between dropout and a fully deterministic model for matrix factorization in which the factors are regularized by the sum of the product of the norms of the columns. While the resulting regularizer is closely related to a variational form of the nuclear norm, suggesting that dropout may limit the size of the factorization, we show that it is possible to trivially lower the objective value by doubling the size of the factorization. We show that this problem is caused by the use of a fixed dropout rate, which motivates the use of a rate that increases with the size of the factorization. Synthetic experiments validate our theoretical findings [Cav+17b].

7. Although dropout is a very effective way of regularizing neural networks and stochastically "dropping out" units with a certain probability discourages over-fitting and improve generalization, one may argue that at the early stages of training, over-redundant correlations between units are unlikely to happen, mainly due to the fact that the network's weights are randomly initialized. Therefore, using a fixed dropout probability during training seems a suboptimal choice as opposed to a more gradual introduction of units suppression.

   To this aim, in [Mor+17], we propose a scheduling in time for the probability of retaining neurons in the network. This induces an adaptive regularization scheme that smoothly increases the difficulty of the optimization problem. This idea of "starting easy" and adaptively increasing the difficulty of the learning problem has its roots in curriculum learning [Ben+09] and allows one to train better models. Indeed, we prove that our optimization strategy implements a very general curriculum scheme, by gradually adding noise to both the input and intermediate feature representations within the network architecture. Experiments on seven image classification datasets and different network architectures show that our method, named curriculum dropout, frequently yields to better generalization and, at worst, performs just as well as the standard dropout method.

1.2 Publications

- **Jacopo Cavazza** and Vittorio Murino - *People Counting by Huber Loss Regression* - Machine Learning for Intelligent Image Processing (ML4IP) - Satellite Workshop of the IEEE International Conference on Computer Vision (ICCVw), oral presentation, December 2015, Santiago, Chile.

- Riccardo Volpi, Andrea Zunino, **Jacopo Cavazza**, Cristina Becchio and Vittorio Murino - *Human Intention Prediction with Unsupervised Feature Learning* - Fifth Italian Workshop on Machine Learning and Data Mining, invited oral presentation, September 2016, Genoa, Italy.
- **Jacopo Cavazza**, Andrea Zunino, Marco San Biagio and Vittorio Murino - *Kernelized Covariance for Action Recognition* - 23rd International Conference on Pattern Recognition (ICPR), December 2016, Cancun, Mexico.
- **Jacopo Cavazza** and Vittorio Murino - *Semi-Supervised Robust Regression with Adaptive Huber Loss* - IEEE Transaction on Cybernetics, May 2017, `under review`.
- Andrea Zunino, **Jacopo Cavazza**, Atesh Koul, Andrea Cavallo, Cristina Becchio and Vittorio Murino - *What Will I Do Next? The Intention from Motion Experiment* - Vision Meets Cognition - Satellite Workshop of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPRw), July 2017, Honolulu, Hawaii, USA.
- **Jacopo Cavazza**, Pietro Morerio and Vittorio Murino - *When Kernel Methods meet Feature Learning: Log-Covariance Network for Action Recognition from Skeletal Data* - Open Domain Action Recognition (ODAR) - Satellite Workshop of the IEEE International Conference on Pattern Recognition (CVPRw), oral presentation, July 2017, Honolulu, Hawaii, USA.
- **Jacopo Cavazza**, Pietro Morerio and Vittorio Murino - *A Compact Kernel Approximation for 3D Action Recognition* - IAPR GIRPR Springer 19th International Conference on Image Analysis and Processing (ICIAP), oral presentation, *Special Mention of Best Paper Award*, September 2017, Catania, Italy.
- Andrea Zunino, **Jacopo Cavazza** and Vittorio Murino - *Revisiting Human Action Recognition: Personalization vs. Generalization* - IAPR GIRPR Springer 19th International Conference on Image Analysis and Processing (ICIAP), September 2017, Catania, Italy.
- Andrea Zunino, **Jacopo Cavazza**, Atesh Koul, Andrea Cavallo, Cristina Becchio and Vittorio Murino - *Predicting Human Intentions from Motion Cues Only: A 2D+3D Fusion Approach* - ACM 25th International Conference on Multimedia (ACMMM), October 2017, spotlight oral presentation, Mountain View, California, USA.
- Pietro Morerio, **Jacopo Cavazza**, Riccardo Volpi, René Vidal and Vittorio Murino - *Curriculum Dropout* - IEEE International Conference on Computer Vision (ICCV), October 2017, Venice, Italy.
- **Jacopo Cavazza**, Benjamin Haeffele, Connor Lane, Pietro Morerio, Vittorio Murino and René Vidal - *Dropout as a Low-Rank Regularizer for Matrix Factorization*, Artificial Intelligence and Statistics (AISTATS), 2018, Lanzarote, Tenerife.
- Pietro Morerio, **Jacopo Cavazza** and Vittorio Murino - *Minimal-Entropy Correlation Alignment for Unsupervised Deep Domain Adaptation*, International Conference on Learning Representations (ICLR), 2018, Vancouver, Canada.

- Andrea Zunino, **Jacopo Cavazza**, Riccardo Volpi, Pietro Morerio, Cristina Becchio and Vittorio Murino - *Predicting Intentions from Motion: the Subject-Adversarial Adaptation Approach*, International Journal of Computer Vision, 2018, `under review`.
- **Jacopo Cavazza**, Pietro Morerio and Vittorio Murino - *Scalable and Compact 3D Action Recognition with Approximated RBF Kernel Machines*, IEEE Transactions on Pattern Analysis and Machine Intelligence (tPAMI), 2018, `under review`. Available as a pre-print on arXiv.

Thesis outline and covered applications.

Due to the variety of different machine learning techniques and computer vision applications investigated in this thesis, for the sake of clarity, its outline is sketched in Figure 1.1 and commented beneath.

In Chapter 2, we will provide necessary background material and revise relevant related works from the literature.

In Chapter 3, we tackle the problem of human action recognition from skeletal data by exploiting newly proposed covariance-based representations [Cav+16] and techniques in which spatio-temporal correlations among variable (joints) are captured.

In Chapter 4, we switch from correlation within different components of the same data representation to correlation between different representations applied to the same data concurrently. To this aim we propose a new robust semi-supervised regression techniques which exploits the Huber loss to automatically spot outliers in the data annotations. A favorable performance is scored in general regression problems, binary classification with noisy labels and crowd counting experiments.

In Chapter 5, we exploit correlations among different visual domains in order to carry out unsupervised adaptation and allow model transfer without performance degradation. A novel geodesic alignment is proposed as well as a new unsupervised cross-validating parameter tuning method based on entropy minimization: overall, our approach score a state-of-the-art performance on benchmark object recognition datasets.

In Chapter 6, we study how to remove redundant correlations among units in an artificial neural network by means of the dropout technique. In a simplified setting, dealing with a matrix factorization model, we draw a principled connection between dropout and Principal Component Analysis (PCA). Inspired by those findings, we propose a generalization of dropout training for deep convolutional neural networks applied to image classification tasks.

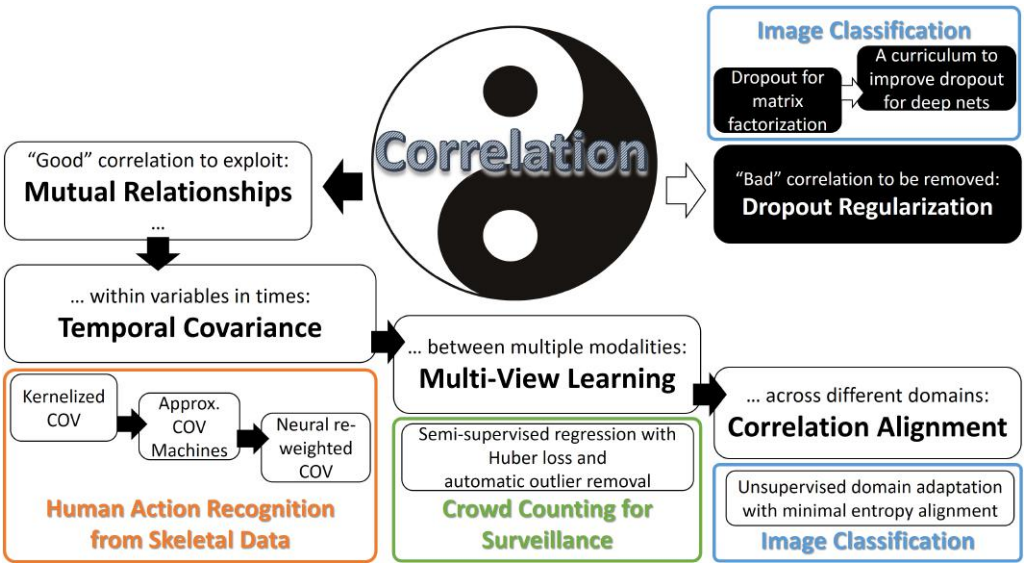


FIGURE 1.1: Thesis outline and covered applications. Correlation can have both positive and negative meanings. *Left and counterclockwise*. Mutual relationships to be captured are an example of provably "good" correlation that one can rely on in visual recognition. As an example of those, we provide different alternative methods in capturing temporal correlation between skeletal joints in order to recognize human actions. Then, we generalize the kind of correlation we capture incrementally. That is, as a first stage, we capture correlations across multiple modalities. By doing so in a robust manner and by embracing semi-supervision with automatic removal of corrupted annotations, we are able to proficiently solve the problem of estimating crowd density in surveillance scenarios. As a second, and last stage, we eventually aim for correlations across datasets (or better, domains) so that we can transfer one model train with a domain A with supervision on a domain B even when no additional annotations are provided. *Top-right*. Correlation can also negatively impact on performance when either the data or the produced feature representations are over-redundant. In such a case the excess of mutual relationships may mislead the learning by adding noisy patterns which are not discriminative for the recognition task. One example of such procedure is the excessive correlation displayed during feature learning by neural networks. With this respect we study an ad-hoc regularization technique, called *dropout*, in order to achieve an improved theoretical understanding of it. Inspired by our findings, we proposed a novel implementation to boost the generalization capabilities of deep nets applied to image classification tasks.

Chapter 2

Background Material and Related Work

In this chapter, we provide some background material to support our dissertation in the remaining part of the thesis. In details, in Section 2.1, we recap the formal definition of correlation as defined in statistics which will be extended in Section 2.2 in its spatio-temporal form. By doing so we will provide the baseline method which will be extended in Chapter 3 with the newly proposed techniques applied to action recognition from skeletal joints.
Section 2.3 provides a broad introduction to multi-view learning and co-training in general: thus, it explains classically adopted tools to combine different feature representations applied on parallel to the same input data. Within this peculiar class of methods, in Chapter 4, we will deploy our proposed robust framework for crowd counting.
In Section 2.4, we introduce the problem of unsupervised domain adaptation that will be the main focus of Chapter 5.
Finally, in Section 2.5, we introduce the regularization scheme for neural networks that is called "dropout" [Hin+12; Sri+14] and which will be extensively investigated in Chapter 6.

All the content of this present chapter is finalized to provide a gentle introduction to the topics covered in the later chapter of the thesis, in addition to revise relevant related works. An expert reader can easily skip this chapter and directly dig into the following chapters where our original contributions are presented. At the same time, single sections of this chapter may be referred as backbone supporting material in the case of basic concepts which are taken for granted in the following pages.

2.1 Correlation: a formal definition

In statistics, dependence or association is any statistical relationship, whether causal or not, between two real, scalar random variables X and Z of the same dimensionality. Despite correlation can refer to any broad class of statistical dependences, though, in common usage, it most often refers to which extent X and Z displays a linear dependence. In formal terms, given n observations $x_1, \dots, x_n \in \mathbb{R}$ of X and n observation $z_1, \dots, z_n \in \mathbb{R}$ of Z, the covariance $\text{cov}(X, Z)$ between X and Z is a scalar number which can be estimated[1] as

$$\text{cov}(X, Z) = \frac{1}{n-1} \sum_{i=1}^{n} (x_i - \bar{x}) \cdot (z_i - \bar{z}), \tag{2.1}$$

where, for notational convenience, $\bar{x}$ and $\bar{z}$ stand for the average of $x_1, \dots, x_n$ and $z_1, \dots, z_n$, respectively. After scaling (2.1) with a division by the product $\mathbb{V}(X)\mathbb{V}(Z)$, the variances of X and Z respectively, one finds the McPearson's correlation index ρ. $\rho \in [-1, 1]$ and spans from a perfect antilinear relationships $Z \propto -X$ when $\rho = -1$ to the direct correlation $Z \propto X$ if $\rho = 1$.

The extension of (2.1) to multiple dimensions is straightforward. Indeed, given n column vectors $\mathbf{x}_1, \dots, \mathbf{x}_n \in \mathbb{R}^d$ sampled from the d-dimensional random vector $\mathbf{X}$, we define the covariance representation (COV) associated to $\mathbf{X}$ as the $d \times d$ matrix $\boldsymbol{\Sigma}(\mathbf{X})$ whose (i, j)-th entry is

$$[\boldsymbol{\Sigma}(\mathbf{X})]_{i,j} = \text{cov}(\mathbf{X}_i, \mathbf{X}_j). \tag{2.2}$$

Or, with the slight abuse of notation which results by denoting $\mathbf{X}$ as the $d \times n$ matrix which stacks by columns $\mathbf{x}_1, \dots, \mathbf{x}_n$, we can compactly write

$$\boldsymbol{\Sigma}(\mathbf{X}) = \frac{1}{n-1} \mathbf{X}\ \mathbf{C}\ \mathbf{X}^\top \tag{2.3}$$

once defined $\mathbf{C}$ as the $n \times n$ centering matrix defines as

$$\mathbf{C}_{ij} = \begin{cases} \dfrac{1-n}{n} & \text{if} \quad i = j \\ -1 & \text{otherwise.} \end{cases} \tag{2.4}$$

As defined in (2.2) (or in its equivalent reformulation (2.3)), $\boldsymbol{\Sigma}(\mathbf{X})$ has a strictly positive trace (since $[\boldsymbol{\Sigma}(\mathbf{X})]_{i,i} = \mathbb{V}(\mathbf{x}_i) > 0$) and is symmetric positive definite [Bha15]. The latter property means that, for each column vector $\mathbf{a} \in \mathbb{R}^d$, it results

$$\mathbf{a}\boldsymbol{\Sigma}(\mathbf{X})\mathbf{a}^\top \geq 0 \tag{2.5}$$

or, equivalently, the eigenvalues of $\boldsymbol{\Sigma}(\mathbf{X})$ are non-negative.

[1] Let us clarify that the rigorous definition of covariance involves the probability distribution p_X and p_Z associated to X and Z. But, in a general applicative case, instead of the true distributions p_X and p_Z, we can access only a limited number of samples $x_1, \dots, x_n \sim p_X$ and $z_1, \dots, z_n \sim p_Z$. This motivates us in circumventing the classical "academical" definition of covariance and directly moving to its *sampling covariance estimator*, which, in this thesis, will be referred as covariance tout-court.

Indeed, originally, Σ was proposed as region descriptor for object recognition and pedestrian detection tasks [Tuz+06a]: in this case $\mathbf{x}_i$ represents a particular image patch. Due to the favorable results achieved, several techniques [Tuz+08; Tos+13; Har+14; Min+14b; Tan+15; Min+16b] have been exploited to perform such tasks directly on the SPD manifold, to which COV matrices belong [Bha15].
Essentially, the whole classification task is resumed in the principled definition of the right distance function in order to compare different COV matrices, as to highlight differences/similarities and feed the classifier on top of such information [Ars+06; Wan+12f; Che+12b; Min+14b; Min+16b].

Grounding on the success on image classification, the covariance representation has been applied to many other recognition task. For instance, for the transfer learning problem of adapting a feature representation learnt on a source domain (with plentiful of annotations) to a target domain which lacks of data and which, therefore, does not allow to perform re-training from scratches. In this case, the adaptation is possible by using the covariance to align second-order statistics of the target domain on the source one [Sun+16; SS16; MM17]. Indeed, fixed a common feature encoding, a covariance matrix Σ_s is computed among the feature vectors computed on the source and then Σ_t is the equivalent computed on the taget domain. Through whitening and re-normalization operations, [Sun+16] is able to align Σ_t on Σ_s. In [SS16], the same idea is implemented with an end-to-end architecture while [MM17] performs a more principled alignment by directly exploiting a Riemannian metric to preserve the SPD structure of Σ_t and Σ_s.

As another paradigm, if we set $\mathbf{x}_1, \dots, \mathbf{x}_n$ to be different temporal acquisitions of the same random vector $\mathbf{X}$ in time, the covariance representation (2.3) actually performs a pooling in time, ultimately achieving a representation which is able to codify the evolution in time of the samples $\mathbf{x}_i$. This is very appealing in the case of action recognition from skeletal data, where $\mathbf{x}_i$ encodes the position of the skeleton at time i by concatenating the $x - y - z$ positions of a certain number of *joints* which represents the intersections between the main bones in the human body. Leveraging on such type of data a number of recent work [Hus+13; Wan+15b; Cav+16; Cav+17c; Cav+17a; Zho+17] has demonstrated the effectiveness of encoding the correlation in time of such skeletal joints in order to achieve state-of-the-art performances.

2.2 Spatio-Temporal Correlation for Human Action Recognition

Human action recognition is a paramount domain in many applicative fields, such as crowd analysis and surveillance, elderly care and autonomous driving vehicles, to name a few. Although the literature has explored a few variants[2]

[2]Readers can refer to Appendix A where we presented in a self-contained manner the following generalization of the original action recognition problem. In fact, not just we want to recognize what is the action displayed but, more challenging, inspect what is the goal which underlies the same action. Leveraging on psychological basis which observes that the same action can be displayed with different intentions, we posit that it's possible to fix one displayed action - say, grasping a bottle - and predict what this action is finalized for - namely, whether

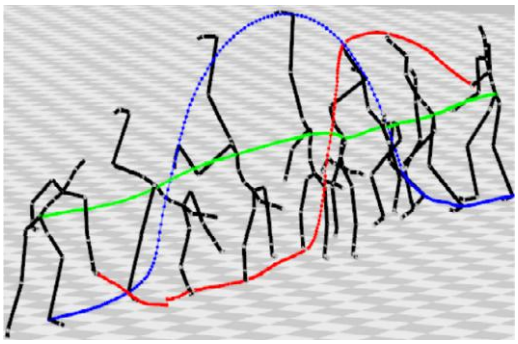


FIGURE 2.1: A cartwheeling action acquired with a motion capture system which is able to track in time the positions of the skeletal joint.

- either including the minimal latency recognition of unfinished actions and the prediction of future intention - as a learning problem, all frameworks can be stated as the classification task of the action label related to a temporal sequence which is trimmed, as to ensure that contains only one single action (such as the cartwheeling act in Fig. 2.1).

Despite the wide interest in video-based approaches, this type of data is intrinsically affected by several issues, *e.g.* privacy, occlusions, light variations and background noise. An effective alternative to deal with these challenges is represented by skeletal based representation. This paradigm relies on theoretical guarantees concerning motion perception. It has in fact been proved by Johansson [Joh73] that the displacement of light sources located on keypoints on the humans' skeleton are enough for the visual system to recognizing the displayed action.

Grounding on that, the evolution of systems which can acquire the skeletal joints nowadays guarantees a reliable estimate of 3D body posture - motion capture, *e.g.* VICON - and a cheap price - depth sensors, *e.g.* Kinect. Additionally, replacing videos with skeletal data does not change the overall general pipeline of action classification: learning/engineering feature representation from trimmed sequences, in order to train a classifier. In practice, for a general action a, skeletal data is acquired in the form of the following multi-dimensional time-series.

$$\mathbf{P}_a = \begin{bmatrix} x_1(t=1) & x_1(t=2) & \dots & x_1(t=T) \\ y_1(t=1) & y_1(t=2) & \dots & y_1(t=T) \\ z_1(t=1) & z_1(t=2) & \dots & z_1(t=T) \\ x_2(t=1) & x_2(t=2) & \dots & x_2(t=T) \\ y_2(t=1) & y_2(t=2) & \dots & y_2(t=T) \\ z_2(t=1) & z_2(t=2) & \dots & z_2(t=T) \\ \vdots & \vdots & \ddots & \vdots \\ x_J(t=1) & x_J(t=2) & \dots & x_J(t=T) \\ y_J(t=1) & y_J(t=2) & \dots & y_J(t=T) \\ z_J(t=1) & z_J(t=2) & \dots & z_J(t=T) \end{bmatrix} \tag{2.6}$$

the bottle has been grasped in order to 1) pour some water into a glass, 2) pass the bottle to another person, 3) drink from the bottle or 4) displace the bottle. In all cases, context is uninformative and the kinematics is the only way to accomplish such prediction. We formulate this problem as Intention from Motion.

where columns correspond to timestamps (from $t = 1$ to $t = T$) and triplets of rows $x_i(t), y_i(t), z_i(t)$ correspond to 3D spatial coordinates of the i-th joint, $i = 1, \dots, J$.

By assuming that the input sequence $\mathbf{P}_a$ is trimmed so that it contains only one action, the ultimate action recognition task is to build some feature representation from $\mathbf{P}_a$ and subsequently train a classifier on top so that, hopefully, an action like the one in Fig. 2.1 can be labeled as "`cartwheeling`".

Clearly, the feature design state is of utmost important to encode $\mathbf{P}_a$ while capturing all the discriminants for classification. Thus, in Sec. 2.2.1, we will overview the most significant works in the literature while shading a particular light on COV-based approaches in Sec. 2.2.2.

2.2.1 Related work in skeletal-based action recognition

In principle, if each frame of a video was vectorized, after stacking all such vectors in a matrix, we could gather the same data structure as in (2.6). This is why, especially in the past year, many algorithms, originally devised for video-based action recognition, have been just brought to the skeletal data paradigm upon minor modifications. Among others, we can mention histogram based representations to perform temporal pooling [YT14a; Eva+14], extraction of local spatio-temporal features from the data [Dev+14; Sei+13; Wen+17], also applying bag-of-words or Fisher vector approaches to aggregate the raw joint representation in a unique action descriptor [Eva+14; Ani+15].

However, the performance of transferring a video-based approach to skeletal data has been proved to be suboptimal with respect to more principled method which endows $\mathbf{P}_a$ in (2.6) with some kind of structure which can be exploited in the classification stage. We call this structured representation a *kernel*. Among the many proposed kernels, we can recall the representation of each joint trajectory as a roto-translation matrix [Vem+14; VC16], which leads to exploit the Lie group and Lie algebra properties of the Special Euclidean group. Alternatively, Hankel matrices [PCSC14; Zha+16a] have been attested to be extremely effective in the field of action recognition from skeletal data, either being paired with Hidden Markov Models [PCSC14] or with a prototype-based nearest neighbor classification on the Riemannian manifold [Zha+16a]. Actually, in both cases of roto-translations and Hankel matrix, countermeasures (such as warping [Vem+14; VC16]) needs to be taken against the following issue: in (2.6), while J is fixed (being an intrinsic parameter of the device used for skeleton's acquisition), T is not, and can in fact changes from action to action (and even among repetition of the same action performed by the same person). Therefore, a pre-processing step (such as warping [Vem+14; VC16]) needs to be applied in order to fix T across instances, since standard methods only deal with fixed-length inputs.

Recently, due to the introduction of the first big dataset for action recognition with skeletal representation [Sha+16], the deep learning paradigm reached a state-of-the-art performance. While leveraging on a structured low-level encoding for the action kinematics - symmetric and positive definite matrices [HG17a] and roto-translations [Hua+17] - classical neural networks module are adapted in order to cope with data which lies on a structured manifold. That

is, [HG17a] performs max pooling on the singular values and, instead of a generic weight matrix, [Hua+17] applies orthonormal linear transformations to preserve the representation on the Lie Group. Alternatively, joint trajectories are used to produce distance maps, then converted into images to fine-tune convolutional neural networks (CNN), which can be therefore applied for 3D action recogniton [Wan+16; Li+17a].
Differently, [Du+15; Sha+16; Liu+16; Liu+17a] applied end-to-end learning from the raw data (2.6) directly. Precisely, [Du+15] proposed a decomposition of the skeleton into torso, legs and arms. Subsequently, a RNN is trained on each part separately and a final fusion is perform just before the classification stage. As to better take advantage of the temporal dimension of the skeletal input data, LSTM architectures have been widely adopted: ranging from the vanilla case of [Sha+16] to the attention mechanisms of either [Liu+16] or [Liu+17a] which better handle missing data.

2.2.2 Covariance-based Representation for Action Recognition from Skeletal Data

In the recent literature on action recognition from skeletal joints [Wan+15b; Cav+16; Zho+17; Cav+17c; Cav+17a], the *covariance representation* (COV) has achieved a predominant role to encode the kinematics.

In formal terms, the covariance matrix $\mathbf{\Sigma}_a$ associated to $\mathbf{P}_a$ is computed according to equations (2.2) and (2.1). Such operation brings a key advantage in adopting a COV-based representation. Indeed, once fixed two joints coordinates - say the x of the i-th joints and the z of the j-th one, we compute the following measure of correlation

$$\mathrm{cov}(x_i(t), z_j(t)) = \frac{1}{T-1}\sum_{t=1}^{T}(x_i(t)-\overline{x_i})\cdot(z_j(t)-\overline{z_j}) \tag{2.7}$$

where T is saturated in the summation and, hence, we are still able to apply the same similarity measure even if T changes from sequence to sequence. In other words, the covariance representation COV is naturally invariant with respect to changes in the speed with which a given action is performed. Additionally, since all pairwise combinations of joints coordinates are taken into account in the computation of $\mathbf{\Sigma}$, the covariance representation can efficiently capture correlation between body parts at fine level.

The latter points justify the empirical success of capturing the correlation in time between skeletal joint positions for the sake of action recognition.

Indeed, [Hus+13] scored outstanding classification results by adopting a temporal pyramid of covariance descriptors, achieving a temporal snippet analysis and therefore capturing well the kinematics. Similarly, [Wan+15b] combined multiple covariance representations (each of them related to a single joint at the time) with a multiple kernel learning approach. In [Kon+16], two covariance-based low-level representations are used to encode the appearance and the kinematics of the skeletal sequences: classification is therefore performed by computing an embedding in a spatio-temporal kernelized feature space and computing a matching score. A state-of-the-art performance is finally achieved

in [Cav+16] by allowing covariance to capture arbitrary, non-linear relationships conveyed by the raw data, technically recovering the kernel trick for covariance estimation. Similarly, a sound performance is achieved in [Cav+17c] by combining the covariance representation with a shallow neural network: differently from the deep (and computational intensive) approaches of [HG17a; Hua+17], [Cav+17c] demonstrated that when the dynamics of the action is modeled through the right structured encoding, there is no need for the classification pipeline to be deep and even a shallow model can match - and even beat - the state-of-the-art. Finally, as opposed to a classical situation in action recognition where gigantic feature vector are produced in order to model the dynamics of an action (*e.g.*, [Kon+16]), kernel approximation has been combined with COV as to achieve scalable classification pipeline by devising a discriminative yet low-dimensional encoding [Cav+17a].

2.3 Correlating multiple *views* for regression tasks and crowd counting

In Sec. 2.1 and in Sec. 2.2, we exploited the notion of covariance as the way to measure how much two (or more) components of the same feature encoding are correlated with each others. In this Section, differently, we will investigate how to model the correlation between alternative feature encodings which may be applied to the same data instance. Indeed, the data may display multiple discriminant characteristics which are potentially very useful to, say, classify those data or, more generally, to accomplish other recognition tasks. In such a case, it is arguably hard to provide one unique encoding to capture all such nuances in the data. Consequently, it is more straightforward to represent the same data with multiple encodings, where each of them is delegated to highligh different discriminative patterns in a separated fashion. However, when multiple parallel encodings are applied to the same data, a new problem arises: how to combine those encodings in one holistic representation. Indeed, a naive approach of just concatenating different encodings is suboptimal due to the fact that 1) some redundancies in the representation may be induced [Vid+16a] and 2) the concatenation may leed to some curse of dimensionality[3] issues which, ultimately, may damage the learning stage [Bis06].

In this Section, we tackle the aforementioned issue by taking advantage of a broad class of machine learning techniques which are termed *multi-view learning*. By defining each *view* as one (out of many) alternative representation with which a data may be encoded, iin contrast to single view learning, multi-view learning jointly optimizes all the functions to directly take advantage of the redundant views of the same input data and improve the learning performance. Therefore, multi-view learning has been receiving increased attention during the latter years (see [Xu+13] for a comprehensive overview) and three main stream paradigms have emerged: subspace learning, multiple kernel learning, and co-training.

[3]The problem of devising algorithms and techniques whose overall behavior is kept unchanged even if scaling up to high dimension is usually known as *course of dimensionality*. Since this is not primarily related with the scope of this thesis, readers can refer to [Bis06, Chapter 1, Section 4.] for a list of evocative intuitions about.

Subspace learning-based approaches aim to obtain a latent subspace shared by multiple views by assuming that the input views are generated from this latent subspace. The dimensionality [Bis06] of the latent subspace is lower than that of any input view, so subspace learning is effective in reducing the "curse of dimensionality". Given this subspace, it is straightforward to conduct the subsequent tasks, such as classification and clustering [refs].

In multiple kernel learning, the separate encodings for each data points are encoded in a separate fashion, separately learning the representation boundaries (*e.g.* , in classification, the ones which divide two classes) which better allows to recognize the data, ultimately envisaging a late fusion of those and performing recognition accordingly [Xu+13].

Antithetically to multiple kernel learning, in co-training the fusion is performed at an early stage so that a unique high-level representation is built across different views and adopted for the ultimate recognition stage.

In this thesis, we will pursue this latter direction and, for the sake of completeness, we will review the most relevant related papers which leverage on co-training or variants in Sec. 2.3.1. Precisely, in this thesis, we will apply co-training-based methods to the problem of estimating the number of pedestrians present in a given environment for video-surveillance purposed (in brief, *crowd counting*). We will explain the rationale behind and present the related literature in Sec. 2.3.2

2.3.1 A literature review on co-training and variants

Co-training [BM98] is one of the earliest schemes for multi-view learning. It trains alternately to maximize the mutual agreement on two distinct views of the unlabeled data. Many variants have since been developed. [NG00] generalized expectation-maximization (EM) by assigning changeable probabilistic labels to unlabeled data. [Mus+02a; Mus+02b; Mus+06] combined active learning with co-training and proposed robust semi-supervised learning algorithms. [Yu+07; Yu+11] developed a Bayesian undirected graphical model for co-training and a novel co-training kernel for Gaussian process classifiers. [WZ10] treated co-training as the combinative label propagation over two views and unified the graph- and disagreement based semi-supervised learning into one framework. [Sin+05] constructed a data-dependent "co-regularization" norm: the resultant reproducing kernel simplified the theoretical analysis and extended the algorithmic scope of co-regularization. [BS04] and [Kum+10; Kum+11] advanced co-training for data clustering and designed effective algorithms for multi-view data. The success of co-training algorithms mainly relies on the three following assumptions

a. **Sufficiency.** Each view is sufficient for classification on its own.

b. **Compatibility.** The target function of both views predict the same labels for co-occurring features with a high probability.

c. **Conditional Independence.** Views are conditionally independent given the label.

Among these assumptions, sufficiency and compatibility are not problematic and are generally satisfied when using features which on purpose models different traits of the data. However, the conditional independence assumption is extremely critical, being in practice replaced by several weaker alternatives [Abn02; Bal+04; WZ07].

2.3.2 An overview on the crowd counting problem

Crowd behaviour analysis has important actual applications both in security or event detection, and has been recently addressed by the computer vision community. In this context, *crowd counting* means estimating the number of people in a certain environment and profiling their dynamics over time. The lack of monitoring in crowding has potentially disastrous consequences. For example, one may remember Hillsborough and Heysel stadium tragedies (in 1985 and 1989, respectively), or the more recent (2010) love parade crowd crush in a music festival in Germany. Due to big amount of videosurveillance data, human control of public gathering is unfeasible. On the other hand, automatic people counting is challenging due to low resolution videos, inter-person occlusions, perspective distortion and more general visual ambiguities related, for example, to light variations [Lei+05],[KC09]. Due to all the aforementioned challenges, for an automatic system is extremely convenient to rely on complementary feature representations, built from the video data, which can tackle those issue separately. Indeed, since achieving invariance towards each of those visual ambiguities is difficult per se, providing one unique feature representation which is able to accommodate all this issue in a unique solution is presumably impossible. Therefore, it is very attractive to exploit methods (such as multi-view learning) which can leverage on multiple, yet complementary, alternative representations of the same data.

In the related literature on crowd counting, a consolidated taxonomy of approaches identifies three main paradigms [Loy+13c]: *counting by detection*, *counting by clustering* and *counting by regression*. In counting by detection, a classifier is trained to learn a model for a single person. This template is convolved with the original image and all the candidate positions for pedestrians are found. After a non maximum suppression, the number of detections will estimate crowd density [Lei+05]. As expected, this type of approaches is sensible to occlusions and deformable part models have been introduced to overcome this issue. For example, encoding shoulder region in a omega-shape pattern is effective in real-word applications [Li+08]. Counting by clustering is based on the extraction of coherent motion pattern from the crowd (*e.g.* with a Kanade-Lucas-Tomasi tracker [RB06]) and a successive clustering phase will give the number of people.

With respect to the previous paradigms, counting by regression is a more straightforward apparatus. Indeed, one can directly estimate the number of people from image features without intermediate steps. Usually, the pipeline starts detecting in each frame a region of interest, while the effects of geometric distortion are removed with an homography [Ma+04]. Some features are extracted from the foreground and a regressor is trained. The works of [Dav+95] and [Ma+04] are based on crowd density modeling assuming a linear-affine relation between the number of people and the edge pixel number, once

perspective distortion is corrected. While [Mar+97] extracted descriptors from mutual occurrences of gray levels, [Cha+08] fixed the most useful features in regression tasks which are mainly based on foreground area, pedestrian edges and texture statistics. Several methods exploited those features, like Bayesian regression models [Cha+09a; CV12] or ridge regression [Che+12a; Che+13a]. A group of recent papers [Tan+11a; Loy+13a] tried to perform manifold learning to exploit geometric inner configuration of input data.

As a recent trend, deep learning approaches have been emerged to a powerful class of algorithms which are able to learn in end-to-end fashion how to map frames from video-surveillance cameras into density map which counts the number of pedestrians in the scene using either convolutional [Wan+15a; Zha+15b; ORLS16; Zha+16b; Boo+16; Sam+17] or recurrent architectures [Ste+16; Han+17; Xio+17]. However, those methods (*e.g.* [Wan+15a; Zha+15b]) usually need bigger amount of annotations if compared to others (*e.g.* [CM15; CM16]): in order to learn the density maps fine-grained annotations are required as to explicitly provide the position in the image of each head of the pedestrians to be counted.

2.4 Correlating visual representations across domains for object categorization

Domain adaptation is a technique which trains a model on a *source* domain with full supervision, then transferring it to a different (but related) *visual* domain: the problem in doing so arises from the fact that source and data distributions are not the same due to general visual ambiguities which translate into a domain shift and dataset bias [TE11]. Consequently, if a model is trained on the source domain and capable of guaranteeing a certain degree of performance on it, when transferred on the target domain, performance degradation occurs due to the aforementioned issue. The final goal of domain adaptation techniques is precisely to recover from such degradation. While doing so, three different scenarios arise with respect to the kind of annotations provided on the target domain during training.

- If the target domain is fully labeled, one refers to this case as *supervised domain adaptation*. It is arguably the easiest case due to the possibility of performing full supervised training on the target and on the source jointly.
- If only a few target instances are annotated, the problem is termed as *semi-supervised domain adaptation*. The problem is slightly more complicated since the transfer learning stage must be partially driven by the raw data and cannot exploit annotations.
- The case of *unsupervised domain adaptation* is arguably the most challenging scenario: in addition to handling a different data distribution with respect to the source one, the target domain provides no annotations and, therefore, adaptation must be carried out at the feature level in such an effective way that domain shift is mitigated in the meantime. In this thesis, we will consider this most challenging case only.

For the problem of (unsupervised) domain adaptation, a first class of methods aims at learning transformations which align feature representations in the source and target sets. For instance, in [Glo+11] auto-encoders are exploited to learn common features. In [Kan+15], a bi-shifting auto-encoder (BSA) is instead intended to shift source domain samples into target ones and, similarly, other methods approach the same problem by means of techniques based on dictionary learning (as in [She+13]). Geodesic methods (such as [GL11; Gon+12] aim at projecting source and target datasets on a common manifold in such a way that the projection already solves the alignment problem. Inspired by the idea of adapting second order statistics between the two domains, [Sun+16; Fer+13] propose a transformation to minimize the distance between the covariances of source and target datasets in order to, ultimately, achieve *correlation alignment*. Due to the well known properties of covariance operators, in some cases [Sun+16], the alignment can be written down in closed-form. But, since the latter operation can be prohibitively expensive in terms of computational cost, [SS16] implements correlation alignment in an end-to-end fashion by means of backpropagation.

A complementary family of approaches exploit the powerful statistical tool of *entropy optimization* in order to carry out adaptation. Indeed, the notion of association [Hae+17c; Hae+17b] is actually implementing explicit entropy minimization [GB04] to align the target to the source embedding by navigating the data manifold by means of closed cyclic paths that interconnect instances belonging to the same objects' classes.

In parallel, there are cases [GL15; Tze+17] where minimax optimization is responsible for doing the following adversarial training. One seeks for feature representations that are effective for the primary visual recognition task being at the same time invariant while changing from source to target. The latter stage is implemented as the attempt of devising a random chance classifier which is asked to detect whether a given feature vector has been computed from a source or target data instance. Therefore, those approaches are implicitly promoting *entropy maximization*[4] at the classifier level.

Finally, entropy regularization is accomplished in [Tze+15a; Car+17; Sai+17] as a complementary step to boost adaptation. Indeed, already established techniques for adaptation such as Batch Normalization [IS15] are applied in low-level layers to align the representations. On top of that, adaptation is refined at the end of the feature hierarchy by introducing a entropy-based regularizer on the target domain based. Practically, the latter exploits network's prediction to generate pseudo-labels [Lee13; Tze+15a]; [Car+17; Sai+17] and compensate for the lack of annotations on the target.

[4] Remember that the distribution that maximes the entropy is the uniform one and, clearly, the latter is the distribution that represents the prediction accomplished by a random chance classifier

2.5 Dropping out redundant correlations in (deep) feature learning

In the previous Sec. 2.1, Sec. 2.2.2 and Sec. 2.3.1, the notion of covariance has been always intended in positive terms which relates to the possibility of accomplishing computer vision tasks by exploiting useful mutual relationships which either different components of the same feature vector or different feature vectors may display.

Differently, in this Section we will investigate a sort of dark side of the notion of correlation, which occurs in the case the input data are excessively correlated, therefore encoding redundant informations. Such excessive overcompleteness is dangerous for recognition purposes since, in this case, correlation does not model useful and discriminant patterns but, on the contrary, constitutes a sort of noise which should be remove from the data.

Different algorithms and techniques have been proposed to perform such cleaning stage or, in more proper terms to decorrelate the input representation, restoring an improved situation where each variable in the data is independent to each other, ultimately devising a more compact, efficient and less ambiguous representation. Surely, the most famous one is Principal Component Analysis (PCA) [Pea01]. This is a statistical procedure that uses an orthogonal transformation to convert a set of observations of possibly correlated variables into a set of values of linearly uncorrelated variables called principal components (or sometimes, principal modes of variation). The number of principal components is less than or equal to the smaller of the number of original variables or the number of observations. This transformation is defined in such a way that the first principal component has the largest possible variance (that is, accounts for as much of the variability in the data as possible), and each succeeding component in turn has the highest variance possible under the constraint that it is orthogonal to the preceding components. Along this lines, many alternative approaches have been proposing by either imposing the constrain of zero correlation in a transformed space (such as in kernelized PCA [Vid+16a] or in Mahalanobis transform [SM09]) or, alternatively, re-defining the notion of independence in alternative manners (for instance, with information theoretic approach, as in Independent Component Analysis [HO00]).

In the recent years, the problem of decorrelating the data representation has emerged as relevant also in the case where such representation is not hand crafted, but, instead, learnt from the data itself. Indeed, since latter stage is generally accomplished by means of artificial neural networks, this due to their hierarchical and compositional structure where units (called *neurons*) are organized into an architecture which is responsible to accomplish the recognition task.

2.5.1 Literature review on dropout

The approach of dropping out units can be traced back to the literature on learning representations from input data corrupted by noise [Bis95; Ben+09; Rif+11]. Indeed, [Bis95] and [Rif+11] provided some theoretical results about the generalization capabilities of discriminative models trained with data which

is artificially corrupted by noise when the latter follows a predefined probability distribution. Also, [Ben+09] explored the possibility of corrupting the data used for training in a progressive models so that easy (*e.g.* , cleaner) examples comes first and hard (*e.g.* , noisy) examples are presented to the classifier for training only afterwards. Such paradigm of creating a curriculum to guide learning from easy to hard example has been applied in many cases: for instance, to image classification tasks [Ben+09; Mor+17].

Since the original formulation [Hin+12; Sri+14], many algorithmic variations of dropout have been proposed. In [Wan+13a], Drop-Connect was proposed as a more general version of Dropout. Instead of directly setting units to zero, only some of the network connections are suppressed. This generalization is proven to be better in performance but slower to train with respect to [Hin+12; Sri+14]. [LBY16] introduce data-dependent and Evolutional-dropout for shallow and deep learning, respectively. These versions are based on sampling neurons form a multinomial distribution with different probabilities for different units. Results show faster training and sometimes better accuracies. In [WM13], as to accelerate dropout, hidden units are dropped out using approximated sampling from a Gaussian distribution: results show that such variant leads to fast convergence without deteriorating the accuracy. [Bay+13] carried out a fine analysis, showing that dropout can be proficiently applied to Recurrent Neural Networks. [WG15] analyzed the effect of dropout on the convolutional layers of a CNN: they define a probabilistic weighted pooling, which effectively acts as a regularizer. [ZZ15] investigated the idea of dropout once applied to matrix factorization while [JF16] introduce a binary belief network which is overlaid on a neural network to selectively suppress hidden units. The two networks are jointly trained, making the overall process more computationally expensive. [Ren+14] proposed to adjust the dropout rate, linearly decreasing the unit suppression rate during training, until the network recovers from overfitting.

Besides all previous works which empirically assess the effectiveness of dropout for (deep) neural network training, a number of paper has tried to provide some theoretical foundations, explaining in which sense dropout acts as a regularizer and why it prevents overfitting to occur. With this respect, [Wag+13] analyzed dropout applied to the logistic loss for fitting (x, y) data pairs where the distribution of y given x is described by a generalized linear model. By means of a Taylor approximation, they show that dropout induces a regularizer that depends on x but not on y. Following on this line of work, [HL15] discussed the mathematical properties of the dropout regularizer (such as non-monotonicity and non-convexity) and derive a sufficient condition to guarantee a unique minimizer for the dropout criterion. [BS13; BS14] considered dropout applied to deep neural networks with sigmoid activations and prove that the weighted geometric mean of all of the sub-networks can be computed with a single forward pass. [Wag+14] investigated the impact of dropout on the generalization error in terms of the bias-variance trade-off. Specifically, they present a theoretical analysis of the benefits related to dropout training under a Poisson topic model assumption in terms of a more favorable bound on the empirical risk minimization. Finally, [GG16] endowed neural networks with a Bayesian framework to handle uncertainty of the network's predictions and investigate the connections between dropout training and inference for deep Gaussian processes.

Chapter 3

Kernelized, Approximated and Data-Driven Spatio-temporal Covariance Machines for 3D action recognition

Human action recognition is a paramount vision task which is defined as the classification of trimmed temporal sequences in which only one action or activity is displayed[1]. Recognizing human actions is crucial for many applicative domain including, but not limiting to, video-surveillance, human-robot interaction, video-games and elderly care. In all those applications, video-based action recognition suffers of visual ambiguities caused by a variety of factors such as light variations from day to night, cloth changes between persons, intra-/inter-person occlusions and background utterance. With this respect, motion capture or depth based approaches are totally able to circumvent those issues and, on top of that, the data acquired through those sensors is very accurate in terms of either spatial or temporal resolution. And, while leveraging on this data modality, recent algorithms' deployment has achieved a great technological level which allows one to reliably estimate skeletal joints which are defined as the intersecting points between two bones of the human skeleton.

Skeleton-based representations for describing human actions root back in the motion perception theory developed in 1950s by Johansons' studies of point-light displacements. In fact, by only observing the temporal trajectories described in time by light points outfitted in correspondence to the skeletal joints, human eyes are totally able to perform activity recognition even if relying on this limited source of information. Arguably, there should be ways to extend

[1] in this Chapter, we will use action or activities interchangeably.

this possibility to artificial eyes of machine learning algorithms applied to computer vision and, during the latter years, skeletal based 3D action recognition has undergone an outstanding development.

Thus, through the already mentioned motion capture or depth commodity systems, trimmed human actions are acquired into the wild and skeletal joints' trajectories are tracked in time (see Figure 3.1 for a complete plot of the acquisition pipeline). If one thinks about, two are the main problems with such representation.

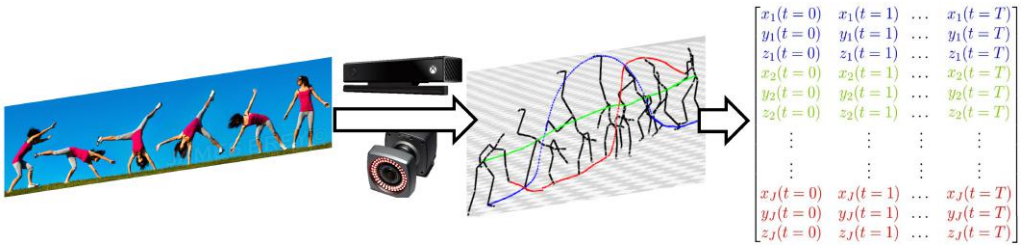


FIGURE 3.1: Skeletal joints acquisition pipeline. We start from a trimmed action (left) where a person performs a cartwheeling actions. Then, through either motion capture or depths sensors (middle) we are able to track in time the J joints, ultimately representing an action as T temporal displacements of skeletons. In practice, when training models for skeletal based action recognition, per each activity, we are asked to process one $3J \times T$ matrix in which triplet of rows correspond to one joints and columns corresponds to timestamps. Note that, usually, J is fixed as soon as we select the acquisition device, while, on the contrary, T varies from instance to instance.

- Despite the number of joints is fixed once we select the device for the acquisition stage – thus the number of trajectory is fixed, however, the temporal duration of those trajectories change from action to action and even from repetition to repetition of the same activity. This is clearly because, even the same activity, can be performed by the same person with different speeds.
- In addition, it can be the case that a certain joint is lost during tracking for a given amount of timestamps. Clearly, the distribution of missing data is extremely sparsified and noisy and, hence, modeling it in an accurate manner is not possible. Then, one should be able to achieve robustness towards such kind of ambiguities.

In this Chapter, we will propose a structured encoding for skeletal joints trajectories in terms of a temporal covariance representation which accumulates in time second order statistics between each possible pairs of joints' coordinates. The resulting encoding is structured as a (positive definite) matrix –SPD - that is able to capture action's kinematics in discriminative way while, at the same time, providing a solution for the aforementioned issues. In fact, covariance representations are naturally invariant towards different execution speeds for actions. Also, by summing in time second order statistics, when some joints are missed for a few, sparse timestamps, the summation is extremely resilient against that.

But, like almost everything in life, covariance representations have also some shortcomings. In fact, they are able to model second-order statistics which originate from linear correlating in time the skeletal joints. Clearly, there might be cases where linear correlations are suboptimal, as displayed in Figure 3.2, and one may ask for the possibility of allowing covariance to model more complex, and thus general, mutual temporal relationships between joints.

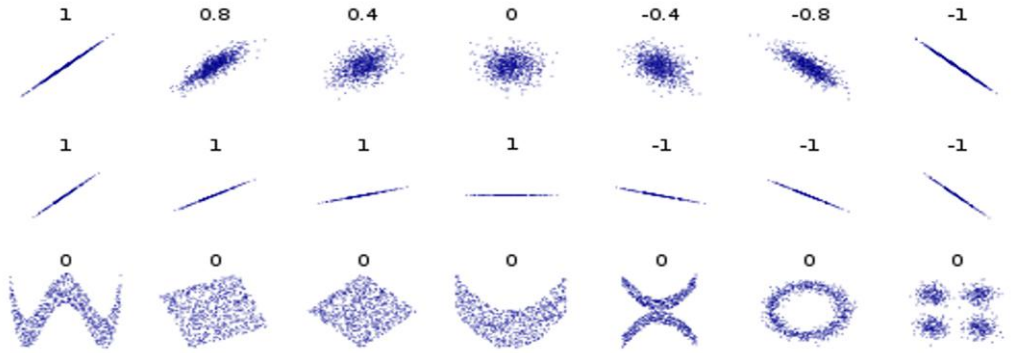


FIGURE 3.2: In this Figure, we represent several sets of bidimensional points x_i, y_i and, for each, we compute the their McPerson's correlation $\rho = \frac{1}{\sigma_x \sigma_y} \sum_i (x_i - x)(y_i - y)$, being x, y the average and σ_x, σ_y the standard deviations of all x_i and y_i, respectively. First row: we can see that a increasing ρ values from -1 to 1 is able to describe well whether the points can be described through a linear model. Second row: ρ is not well sensitive with respect to little changes in the orientations of the points, as soon as the lie on a straight line. Third row: despite all these set present an interesting shaped structure, ρ is not able to appreciate it and, on the contrary, all such sets are indistinguishable for such indicator.

At the same time, due to the fact that covariance representations are SPD operators, it has been shown that conventional classification pipelines are suboptimal with respect to alternative manifold-aware classifier. Indeed, the latter ones are able to achieve classification on the SPD manifold directly, computing similarities between actions – represented as covariance matrices – by taking into account to the inner curvature of the feature space.

Finally, even if covariance representations are extremely rich in capturing all possible correlations between pairs of joints trajectories, maybe only *some* of those are really crucial for the actual recognition and, complementarily, other may be misleading since common within many actions - for instance, basketball and tennis playing are very common in terms of leg movements (both requires running) and can be differentiated by looking at the movements of the upper part of the human body.

In this chapter, we tackle all the aforementioned problems through the following main contributions.

1. We aim at increasing the descriptive power of the covariance matrix, limited in capturing linear mutual dependencies between variables only. We present a rigorous and principled mathematical pipeline to recover the kernel trick for computing the covariance matrix, enhancing it to model more complex, non-linear relationships conveyed by the raw data.

To this end, we propose ***Kernelized-COV***, which generalizes the original covariance representation without compromising the efficiency of the computation. In the experiments, we validate the proposed framework against many previous approaches in the literature, scoring on par or superior with respect to the state of the art on benchmark datasets for 3D action recognition – Section 3.1

2. 3D action recognition was shown to benefit from a covariance representation of the input data (joint 3D positions). A kernel machine feed with such feature is an effective paradigm for 3D action recognition, yielding state-of-the-art results. Yet, the whole framework is affected by the well-known scalability issue. In fact, in general, the kernel function has to be evaluated for all pairs of instances inducing a Gram matrix whose complexity is quadratic in the number of samples. In this work we reduce such complexity to be linear by proposing a novel and explicit feature map to approximate the kernel function. This allows to train a linear classifier with an explicit feature encoding, which implicitly implements a Log-Euclidean machine in a scalable fashion. Not only we prove that the proposed approximation is unbiased, but also we work out an explicit strong bound for its variance, attesting a theoretical superiority of our approach with respect to existing ones. Experimentally, we verify that our representation provides a compact encoding and outperforms other approximation schemes on a number of publicly available benchmark datasets for 3D action recognition. – Section 3.2.

3. We learn how to weight covariance representation in order to highlight those joints' trajectories which are more representative with respect to each single action class in favor of the remaining one. This is done by intertwining covariance representations with a neural network formalism which is able to discriminatively enhance covariance representation. Differently from available deep learning architecture, since we posit that covariance representation is a powerful tool to encode kinematics, we provide experimental evidences to support our claim that, when the dynamics is properly captures there is no need for the architecture to be deep nor recurrent in order to score favorably in terms of (improvements) over state-of-the-art methods for 3D human action recognition – Section 3.3

In the following three sections, we will investigate each of those three problems separately and, afterwards, we will draw conclusions in Section 3.4.

3.1 Kernelizing temporal covariance

At an arbitrary timestamp t, a generic MoCap system represents the body of a human agent as the collection $\mathbf{x}(t) \in \mathbb{R}^{3n}$ of the three-dimensional locations $\mathbf{x}_1(t), \dots, \mathbf{x}_n(t)$ of n joints/markers positions, being $\mathbf{x}_i(t) = [x_i(t), y_i(t), z_i(t)]^\top \in \mathbb{R}^3$ the x, y and z coordinates for $i = 1, \dots, n$. In order to quantify how much any pair of the coordinates mutually change in time, the notion of covariance is classically exploited in statistics [Ham94]. However, it cannot be computed in absence of a known distribution for the probability according to which the samples $\mathbf{x}(t)$ are drawn. However, this assumption is seldom verified in

real cases and, as an alternative, the sampling covariance matrix $\widehat{\mathbb{S}}$ is usually exploited: this is due to the fact that it is an unbiased estimator of the original covariance[2] and can be computed using a finite number of samples $\mathbf{x}(t)$, $t = 1, \dots, T$, only. Precisely, it is defined as

$$\widehat{\mathbb{S}}(\mathbf{X}) = \frac{1}{T-1} \sum_{t=1}^{T} (\mathbf{x}(t) - \boldsymbol{\mu})(\mathbf{x}(t) - \boldsymbol{\mu})^{\top}, \tag{3.1}$$

where $\mathbf{X}$ represents the $3n \times T$ data matrix which stacks by columns all the temporal acquisitions $\mathbf{x}(1), \dots, \mathbf{x}(T)$, whose average is denoted by $\boldsymbol{\mu}$.

Theorem 1. *In matrix notation,* (3.1) *becomes*

$$\widehat{\mathbb{S}}(\mathbf{X}) = \mathbf{X}\mathbf{P}\mathbf{X}^{\top}, \tag{3.2}$$

once defined $\mathbf{P}$ *as the* $T \times T$ *matrix whose* (s,t)*-th entry is*

$$\mathbf{P}_{ss} = \frac{1}{T} \qquad \textit{and} \qquad \mathbf{P}_{st} = -\frac{1}{T^2 - T} \quad \textit{if } s \neq t. \tag{3.3}$$

Proof. Let us define with s_{ij} the generic entry of $\widehat{\mathbb{S}}(\mathbf{X})$ of row i and column j. It results

$$\begin{aligned} s_{ij} &= \frac{1}{T-1} \sum_{t=1}^{T} \left(\mathbf{X}_{it} - \frac{1}{T} \sum_{s=1}^{T} \mathbf{X}_{is} \right) \left(\mathbf{X}_{jt} - \frac{1}{T} \sum_{r=1}^{T} \mathbf{X}_{jr} \right) \\ &= \frac{1}{T-1} \sum_{t=1}^{T} \mathbf{X}_{it}\mathbf{X}_{jt} - \frac{1}{T(T-1)} \sum_{t=1}^{T} \sum_{r=1}^{T} \mathbf{X}_{it}\mathbf{X}_{jr} \\ &\quad - \frac{1}{T(T-1)} \sum_{t=1}^{T} \sum_{s=1}^{T} \mathbf{X}_{jt}\mathbf{X}_{is} + \frac{1}{T^2(T-1)} \sum_{t=1}^{T} \sum_{s=1}^{T} \sum_{r=1}^{T} \mathbf{X}_{is}\mathbf{X}_{jr} \end{aligned} \tag{3.4}$$

In the last summation in the right side of (3.4) there is no addend which depends on t, thus

$$\begin{aligned} s_{ij} &= \frac{1}{T-1} \sum_{t=1}^{T} \mathbf{X}_{it}\mathbf{X}_{jt} - \frac{1}{T(T-1)} \sum_{t=1}^{T} \sum_{r=1}^{T} \mathbf{X}_{it}\mathbf{X}_{jr} \\ &\quad - \frac{1}{T(T-1)} \sum_{t=1}^{T} \sum_{s=1}^{T} \mathbf{X}_{jt}\mathbf{X}_{is} + \frac{1}{T(T-1)} \sum_{s=1}^{T} \sum_{r=1}^{T} \mathbf{X}_{is}\mathbf{X}_{jr} \end{aligned} \tag{3.5}$$

since the summation over t counts T elements and we also simplified with the T in the denominator. In the right side of (3.5) the second and fourth addends are equal in magnitude and opposite in sign: this follows by modifying the summation index in the fourth addends according to the transformation $s \mapsto t$. Therefore

$$s_{ij} = \frac{1}{T-1} \sum_{t=1}^{T} \mathbf{X}_{it}\mathbf{X}_{jt} - \frac{1}{T(T-1)} \sum_{t=1}^{T} \sum_{s=1}^{T} \mathbf{X}_{jt}\mathbf{X}_{is}.$$

[2] For convenience, in the following, we will concisely refer to the estimator $\widehat{\mathbb{S}}$ as the covariance itself, omitting the "sampling" attribute.

We can exploit the properties of Kronecker symbol, consequently obtaining

$$\begin{aligned} s_{ij} &= \frac{1}{T-1}\sum_{t=1}^{T}\sum_{s=1}^{T}\mathbf{X}_{is}\delta_{st}\mathbf{X}_{jt} - \frac{1}{T(T-1)}\sum_{t=1}^{T}\sum_{s=1}^{T}\mathbf{X}_{jt}\mathbf{X}_{is} \\ &= \sum_{s=1}^{T}\sum_{t=1}^{T}\mathbf{X}_{is}\left(\frac{\delta_{st}}{T-1} - \frac{1}{T(T-1)}\right)\mathbf{X}_{jt} \end{aligned} \tag{3.6}$$

From (3.6), for every $s,t = 1,\dots,T$ the definition $\mathbf{P}_{st}$ according to the first equality of (3.3) ensures that the second one is immediately verified (this is easily checked with a few algebra). Thus, (3.6) rewrites

$$s_{ij} = \sum_{s=1}^{T}\sum_{t=1}^{T}\mathbf{X}_{is}\mathbf{P}_{st}\mathbf{X}_{jt} = \sum_{s=1}^{T}\sum_{t=1}^{T}\mathbf{X}_{is}\mathbf{P}_{st}\left(\mathbf{X}^{\top}\right)_{tj} \tag{3.7}$$

which produces the thesis thanks to the formal definition of the row-by-column matrix product and the arbitrary indexes i and j considered. □

Observation. In the present thesis, as a working assumption, we will assume that the symmetric and positive definite matrices (SPD) that we will handle are full rank. Since we focus our attention to the case of computing temporal covariance of skeletal joints, we can easily satisfy the full rank assumption in cases of longer actions, where the number of temporal acquisitions are more than the number of skeletal joints. Since the number of skeletal joints is never bigger than 20-30, we only require the action length to be at minimum than 20-30 timestamps. Although the previous case is almost always satisfied, there exists cases of instantaneous actions where the full rank assumption does not hold. In order to circumvent that issue, we will make sure that the full rank assumption is satisfied by applying *regularization* to the spectrum. That is, we add a small ϵ value to the singular values of our SPD operators as to make sure that none of them is zero. In practice, we will use $\epsilon = 10^{-5}$.

The usage of the covariance $\widehat{\mathbb{S}}$ to produce descriptors for classification tasks has been intensively studied [Tuz+06b; Pan+08; Tos+13; Bia+13; Min+14b; Min+16b; Roz+16]. In particular, [Tuz+06b] proposed patch-specific covariance descriptors, efficiently computed with integral images. Other approaches rely on covariance to systematically encode mutual relationships inside the data and such idea was applied to many different applications such as face recognition [Pan+08], person identification [Tos+13] and more general classification tasks [Bia+13]. Further, covariance was proposed to measure similarities across data samples [Bia+13].

This latter direction actually grounds on the mathematical properties of positive definite matrices, exploiting Riemannian metrics on manifold for image classification: once moved from a finite to an infinite dimensional space, the performance enhances [Min+14b; Har+14] and only recently deep learning approaches have shown to be superior. However, one of the main limitation related to covariance matrix is that it only enables to capture linear inter-relationships [Ham94]. For instance, principal component analysis actually exploits a covariance matrix to remove linear correlation of data points [Bis06]. Among the attempts for modeling more complicated relationships, additional statistics, such as entropy and mutual information [Bia+13], and kernels [Wan+15b] have

been adopted. As a different paradigm, one can model non-linear behaviors by preliminary applying a preprocessing step and encode raw data by means of a transformation which increases the feature space. For instance, [San+13] applied such idea for spatial and temporal derivatives for gesture recognition, [Bia+13] considered both different color spaces and edge detectors for image classification, and [Tos+13] used filter bank responses as features to estimate head orientation. In this latter approach, once defined the feature map Φ and the transformed data matrix $\mathbf{\Phi}(\mathbf{X})$ whose t-th column is $\Phi(\mathbf{x}(t))$, the covariance (3.2) is now expressed by

$$\widehat{\mathbb{S}}(\mathbf{\Phi}(\mathbf{X})) = \mathbf{\Phi}(\mathbf{X})\mathbf{P}\mathbf{\Phi}(\mathbf{X})^{\top}. \tag{3.8}$$

Despite $\widehat{\mathbb{S}}(\mathbf{\Phi}(\mathbf{X}))$ is able to capture general relationships embedded in the raw data $\mathbf{X}$, the main bottleneck with (3.8) is the requirement of explicit computation for $\mathbf{\Phi}(\mathbf{X})$. Indeed, due to feature space augmentation performed by Φ, the higher dimensionality of such a matrix is more demanding in terms of both storage and computational cost required to calculate (3.8) instead of (3.2). Additionally, although infinite feature spaces are common for many classes of feature maps (*e.g.*, the one corresponding to a Gaussian kernel), this case has to be excluded in (3.8) since $\mathbf{\Phi}(\mathbf{X})$ is infinite dimensional and therefore impossible to compute exactly. In the following Section, we will face the problem of obtaining $\widehat{\mathbb{S}}$ without involving $\mathbf{\Phi}(\mathbf{X})$.

3.1.1 Recovering the kernel trick for covariance representations

Leveraging on the theory of kernel methods [SS02], every symmetric and positive definite kernel function $k\colon \mathbb{R}^{3n} \times \mathbb{R}^{3n} \to \mathbb{R}$ can be expressed as

$$k(\mathbf{x}, \mathbf{z}) = \langle \Phi(\mathbf{x}), \Phi(\mathbf{z}) \rangle_{\mathcal{H}}, \tag{3.9}$$

where the inner product is computed in the Hilbert space $\mathcal{H}$ which defines the range of the feature map $\Phi\colon \mathbb{R}^{3n} \to \mathcal{H}$. In (3.9), the kernel trick [SS02] replaces the arbitrary relationships in the original data space with a linear reformulation in $\mathcal{H}$: most importantly, Φ can be actually skipped, since only requiring the computation of the kernel k (*e.g.*, this happens for support vector machines [Bis06]). In our case, we will employ k to obtain the representation $\widehat{\mathbb{S}}(k)$, equivalent to (3.8), that is $\widehat{\mathbb{S}}(k) = \widehat{\mathbb{S}}(\Phi(\mathbf{X}))$, while also skipping the computation of Φ. The following statement moves the first step in this direction.

Lemma 1. *Assume that there exist* $\mathbf{h}_j \in \mathbb{R}^{3n}$ *such that* $\Phi(\mathbf{h}_j) = \mathbf{e}_j$ *for every* $j = 1, \dots, \dim(\mathcal{H})$, *being* $\mathbf{e}_j$ *the unitary element of the canonical base of* $\mathcal{H}$ *as a vectorial space. Then, there exists a* $\dim(\mathcal{H}) \times T$ *matrix* $\mathbf{K}[\mathbf{X}, \mathbf{h}]$, *depending only on the kernel* k, *the data* $\mathbf{X}$ *and* $\mathbf{h}_j$, *such that, if we define* $\widehat{\mathbb{S}}(k) = \mathbf{K}[\mathbf{X}, \mathbf{h}]\mathbf{P}\mathbf{K}[\mathbf{X}, \mathbf{h}]^{\top}$, *we get* $\widehat{\mathbb{S}}(k) = \widehat{\mathbb{S}}(\Phi(\mathbf{X}))$.

Proof. Using (3.8), the (i,j)-th entry of $\widehat{\mathbb{S}}(\Phi(\mathbf{X}))$ rewrites

$$\widehat{\mathbb{S}}_{ij}(\Phi(\mathbf{X})) = \sum_{s,t=1}^{T} \langle \Phi(\mathbf{x}(s)), \mathbf{e}_i \rangle_{\mathcal{H}} \mathbf{P}_{st} \langle \Phi(\mathbf{x}(t)), \mathbf{e}_j \rangle_{\mathcal{H}}. \tag{3.10}$$

In (3.10), once exploited the assumption that $\Phi(\mathbf{h}_j) = \mathbf{e}_j$, for some $\mathbf{h}_j$, we can define the $\dim(\mathcal{H}) \times T$ matrix $\mathbf{K}[\mathbf{X}, \mathbf{h}]$ whose (i, s)-th entry $k(\mathbf{x}(s), \mathbf{h}_i)$ is $\langle \Phi(\mathbf{x}(s)), \mathbf{e}_i \rangle_{\mathcal{H}} = \langle \Phi(\mathbf{x}(s)), \Phi(\mathbf{h}_i) \rangle_{\mathcal{H}}$ and consequently we deduce

$$\widehat{\mathbb{S}}(k) = \mathbf{K}[\mathbf{X}, \mathbf{h}]\mathbf{P}\mathbf{K}[\mathbf{X}, \mathbf{h}]^\top = \widehat{\mathbb{S}}(\Phi(\mathbf{X})), \tag{3.11}$$

which proves the thesis. □

Lemma 1 certifies that we are able to compute the covariance in terms of the sole kernel k. However, some issues pertain to the practical feasibility of the assumption

$$\Phi(\mathbf{h}_j) = \mathbf{e}_j, \tag{3.12}$$

for any j, which is nevertheless fundamental for our purposes.

Actually, (3.12) is quite restrictive since the range of Φ is forced to contain the whole canonical base of $\mathcal{H}$. For instance, if $\mathcal{H} = \mathbb{R}^M$, (3.12) consists in a set of M equations that have to be solved in an M-dimensional space and, even if we assume that $\Phi(\mathbf{x}) = \mathbf{x}$, the resulting linear system can be either undetermined or impossible. Clearly, in case of a more general shape for Φ, it is not trivial to check whether the assumption (3.12) is verified. Hence, it seems natural to opt for a different feature map, which can replace Φ in generating the kernel function k, also satisfying (3.12). Thus, in the rest of this Chapter, we will focus on a specific class of stochastic feature maps $\boldsymbol{\Psi}$, actually fulfilling hypothesis (3.12), so that the induced linear kernel approximates k in a both stochastic and analytical sense. Therefore, we select the family of functions

$$k(\mathbf{x}, \mathbf{z}) = \sum_{\ell=0}^{\infty} a_\ell \langle \mathbf{x}, \mathbf{z} \rangle^\ell \tag{3.13}$$

where the dot product $\langle \mathbf{x}, \mathbf{z} \rangle$ is computed in $\mathbb{R}^{3n}$ and $a_\ell \geq 0$ for any ℓ. It is worth nothing that, due to the non-negativeness of these coefficients, since a linear combination of kernels is still positive definite, then (3.13) admits the representation (3.9). Also, (3.13) covers both finite and infinite linear combinations and therefore is comprehensive of a broad class of kernel functions. For instance, it is easily checked that (3.13) generalizes both the polynomial kernel $k(\mathbf{x}, \mathbf{z}) = \langle \mathbf{x}, \mathbf{z} \rangle^\ell + a_0$ and the exponential-dot product kernel $k(\mathbf{x}, \mathbf{z}) = \exp\left(\frac{\langle \mathbf{x}, \mathbf{z} \rangle}{\sigma^2}\right)$, $\sigma > 0$. In this setting, we now introduce the following lemma which gives the fundamental tool to construct $\boldsymbol{\Psi}$.

Lemma 2. *Let $\boldsymbol{\omega} = [\omega_1, \ldots, \omega_{3n}]$ a collection of $3n$ independent samples jointly distributed as a mixture of discrete Dirac's deltas and define $\psi(\mathbf{x}) = \langle \boldsymbol{\omega}, \mathbf{x} \rangle$. Then, the expectation of $\psi(\mathbf{x})\psi(\mathbf{z})$ under the distribution of $\boldsymbol{\omega}$ is*

$$\mathbb{E}_{\boldsymbol{\omega}}[\psi(\mathbf{x})\psi(\mathbf{z})] = \langle \mathbf{x}, \mathbf{z} \rangle. \tag{3.14}$$

Proof. Using the definition of ψ, the property of the mixture of Dirac's delta distribution and the linearity of the expectation $\mathbb{E}_{\boldsymbol{\omega}}$, the thesis comes after the

following chain of equivalences

$$\mathbb{E}_{\omega}[\psi(\mathbf{x})\psi(\mathbf{z})] = \mathbb{E}_{\omega}[\langle\omega,\mathbf{x}\rangle\langle\omega,\mathbf{z}\rangle] = \mathbb{E}_{\omega}\left[\sum_{i,j=1}^{3n}\omega_i\omega_j\mathbf{x}_i\mathbf{z}_j\right]$$
$$= \sum_{i,j=1}^{3n}\mathbb{E}_{\omega}[\omega_i\omega_j]\mathbf{x}_i\mathbf{z}_j = \sum_{i,j=1}^{3n}\delta_{ij}\mathbf{x}_i\mathbf{z}_j = \langle\mathbf{x},\mathbf{z}\rangle,$$

where δ_{ij} denotes the Kronecker symbol. □

Once sampled a random number $N \in \mathbb{N}$ with probability $\frac{1}{p^{N+1}}$, define $\mathbf{\Psi}(\mathbf{x}) = \frac{1}{\sqrt{M}}[\Psi_1(\mathbf{x}),\dots,\Psi_M(\mathbf{x})]$ where $\Psi_1,\dots,\Psi_M$ are all identical copies of the function

$$\mathbf{x} \longmapsto \sqrt{a_N p^{N+1}}\prod_{j=1}^{N}\langle\omega_j,\mathbf{x}\rangle, \tag{3.15}$$

where $\omega_1,\dots,\omega_N$ are independently distributed according to ω. Equation (3.15) and Lemma 2 allow to extend to our case [KK12, Lemma 7], which states that the linear kernel $\langle\mathbf{\Psi}(\mathbf{x}),\mathbf{\Psi}(\mathbf{z})\rangle$ obtained through $\mathbf{\Psi}$ is an unbiased estimator of the original function $k(\mathbf{x},\mathbf{z})$. Similarly, using the same arguments of Section 4.1 in [KK12], we obtain that $\langle\mathbf{\Psi}(\mathbf{x}),\mathbf{\Psi}(\mathbf{z})\rangle \approx k(\mathbf{x},\mathbf{z})$ uniformly over any compact set of $\mathbb{R}^{3n}$.

Since we proved that $\mathbf{\Psi}$ approximates the kernel k in the sense explained above, the final stage is solving the issue related to (3.12).

Proposition 1. *The map $\mathbf{\Psi}$ satisfies the assumption (3.12), that is, for every i $1,\dots,M$, it results*

$$\frac{1}{\sqrt{M}}[\Psi_1(\mathbf{h}_i),\dots,\Psi_M(\mathbf{h}_i)] = \mathbf{e}_i. \tag{3.16}$$

Proof. The relationship (3.16) displays a system of equations, stochastically dependent on the randomness of $\mathbf{\Psi}$. Actually, in our case, it is enough to solve the system (3.16) and prove the existence of $\mathbf{h}_1,\dots,\mathbf{h}_M$ under a specific realization of N and ω, the two sources of randomness in $\mathbf{\Psi}$. In other words, we can solve (3.16) in a maximum likelihood sense by considering the samples of N and ω which verify (3.16) with probability 1. Thus, we use a prior on N so that $N = 1$ and, once absorbed into $\mathbf{h}_i$ all the multiplicative constant defining $\mathbf{\Psi}$, then (3.16) becomes

$$[\langle\omega_1,\mathbf{h}_i\rangle,\dots,\langle\omega_M,\mathbf{h}_i\rangle] = \mathbf{e}_i, \quad i = 1,\dots,M. \tag{3.17}$$

Precisely, (3.17) is a linear system of size M in the M unknowns $\mathbf{h}_i$. If we then assume that the Dirac delta distribution of ω_j is concentrated in j with probability 1, (3.17) is solvable if and only if $\langle\omega_j,\mathbf{h}_i\rangle = \delta_{ij}$ for any $i,j = 1,\dots,M$. This is actually verified once chosen $\mathbf{h}_i$ to be the i-th element of the orthonormal basis of $\mathbb{R}^{3n}$. □

With Proposition 1, all issues related to the computability for $\widehat{\mathbb{S}}(k)$ is solved. Additionally, one can also easily understand that, with the previous choice of

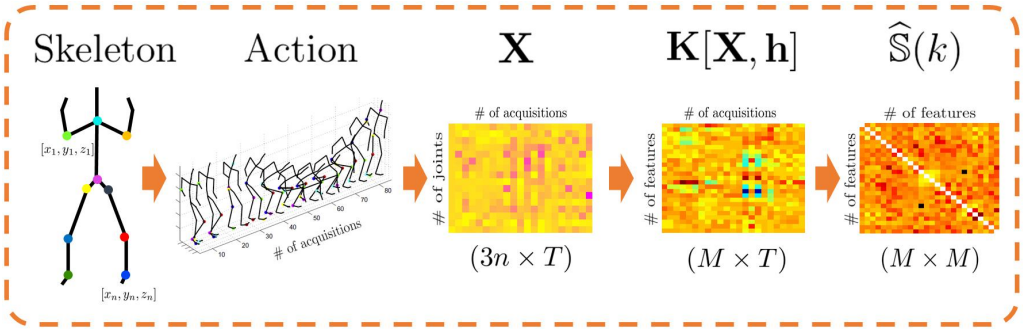


FIGURE 3.3: Overview of the proposed encoding. From the human skeleton, for each action, we extract MoCap data. The latter are represented through the matrix $\mathbf{X}$ which collects the three-dimensional coordinates, referring to the n joints, acquired during T successive instants. A kernel encoding is performed by means of the Gram matrix $\mathbf{K}[\mathbf{X}, \mathbf{h}]$, which is finally used to compute the kernelized covariance $\widehat{\mathbb{S}}(k)$.

Algorithm 1: Pseudo-code of our paradigm.

Input: Set of actions, kernel function k as in (3.13).

Output: Kernelized covariance matrix $\widehat{\mathbb{S}}(k)$ (used as input to a classifier).

Procedure:

For each action, extract the data matrix $\mathbf{X}$ collecting all the T temporal acquisitions $\mathbf{x}(1), \ldots, \mathbf{x}(T)$, each of them encoding the 3D coordinates of the n joints;

For each data matrix $\mathbf{X}$, select $\mathbf{h}_1, \ldots, \mathbf{h}_M$ as in Proposition 1 and compute the Gram matrix $\mathbf{K}[\mathbf{X}, \mathbf{h}]$ according to Lemma 1;

Compute the linear operator $\mathbf{P}$ defined in (3.3);

By means of $\mathbf{K}[\mathbf{X}, \mathbf{h}]$ and $\mathbf{P}$, computed in the previous steps, use (3.11) to calculate the kernelized covariance $\widehat{\mathbb{S}}(k)$;

$\mathbf{h}_i$, once selected a linear kernel $k(\mathbf{x}, \mathbf{z}) = \langle \mathbf{x}, \mathbf{z} \rangle$, then $\widehat{\mathbb{S}}(k)$ is equal to the $\widehat{\mathbb{S}}(\mathbf{X})$, so that the classical covariance is a particular case of our framework.

The theoretical discussion leads to derive Algorithm 1 and to apply the proposed kernelized covariance for the task of action and activity recognition. For a better understanding, we also visualize such pipeline in Figure 3.3.

Computational cost. The complexity of our trial-specific kernelized covariance is $O(M^2T^2)$. Thus, differently from previous approaches [Bia+13; Jay+13; Min+14b; Har+14], the proposed framework is very efficient if compared to the cubic complexity of methods like [Jay+13] which require eigen-decomposition. Under a mathematical point of view, our kernelized covariance is a natural generalization of the classical covariance matrix, which can be retrieved as a particular case in our paradigm once fixed the kernel function (3.13) to be a linear one. On the other hand, the computational cost still remains the same if compared with the classical covariance descriptor.

Method	MSR-Action3D	MSR-Daily-Activity
Region-COV [Tuz+06b]	74.0%	85.0%
Hierarchy of COVs [Hus+13]	90.5%	-
COV-$J_{\mathcal{H}}$-SVM [Har+14]	80.4%	75.5%
Ker-RP-POL [Wan+15b]	96.2%	**96.9%**
Ker-RP-RBF [Wan+15b]	**96.9%**	96.3%
Kernelized-COV (proposed)	96.2%	96.3%

Method	MSRC-Kinect12	HDM-05
Region-COV [Tuz+06b]	89.2%	91.5%
Hierarchy of COVs [Hus+13]	91.7%	-
COV-$J_{\mathcal{H}}$-SVM [Har+14]	89.2%	82.5%
Ker-RP-POL [Wan+15b]	90.5%	93.6%
Ker-RP-RBF [Wan+15b]	92.3%	96.8%
Kernelized-COV (proposed)	**95.0%**	**98.1%**

TABLE 3.1: Comparative performance of the proposed kernelized-COV benchmarking previous methods in the literature [Tuz+06b; Hus+13; Har+14] based on covariance representations. Best results in bold.

In this section, we present the experimental results obtained with our *Kernelized-COV* method on different publicly available MoCap datasets for action recognition. Precisely, the following algorithms were compared in our experiments: *Region-COV* [Tuz+06b] (covariance region descriptor), temporal pyramid of covariance descriptors (*Hierarchy of COVs*) [Hus+13] and, finally, an infinite covariance operator which exploits Bregman divergence, namely *COV-$J_{\mathcal{H}}$-SVM* [Har+14]. Furthermore, we also report the comparison against the recent state-of-the-art methods, namely *Ker-RP-POL* and *Ker-RP-RBF* [Wan+15b].

In all the experiments, we followed [Wan+15b] in performing SVM classification by means of a global log-Euclidean kernel applied upon Gram matrices, directly computed over joints coordinates, encoding each single trial. Nevertheless, differently from [Wan+15b], in order to represent each multivariate time series of joints trajectories, the data encoding of any trial was realized through our kernelized covariance matrix $\widehat{\mathbb{S}}(k)$, where k is the exponential-dot product kernel. For a fair comparison, our kernelization was plugged into the publicly available code[3] and, for classification, we used the *SVM and Kernel Methods Matlab Toolbox*[4] using the wrapper directly provided by the authors. Finally, we fixed $M = 3n$ and, as done by [Wan+15b], the kernel parameter $\sigma > 0$ is chosen by cross validation. A visualization of the adopted classification pipeline is available in Figure 3.4

In all the experiments, we only used the 3D skeleton coordinates available in the following datasets:

- MSR-Action3D [Li+10b], where there are 20 classes of mostly sport-related action (*e.g.*, *jogging* or *tennis-serve*) involving 10 subjects. Since each subject performs each action 2 or 3 times, the overall number of trials is 567. For each of them, Kinect sensor is used to acquire depth

[3] http://www.uow.edu.au/~leiw/

[4] http://asi.insa-rouen.fr/enseignants/~arakoto/toolbox/index.html

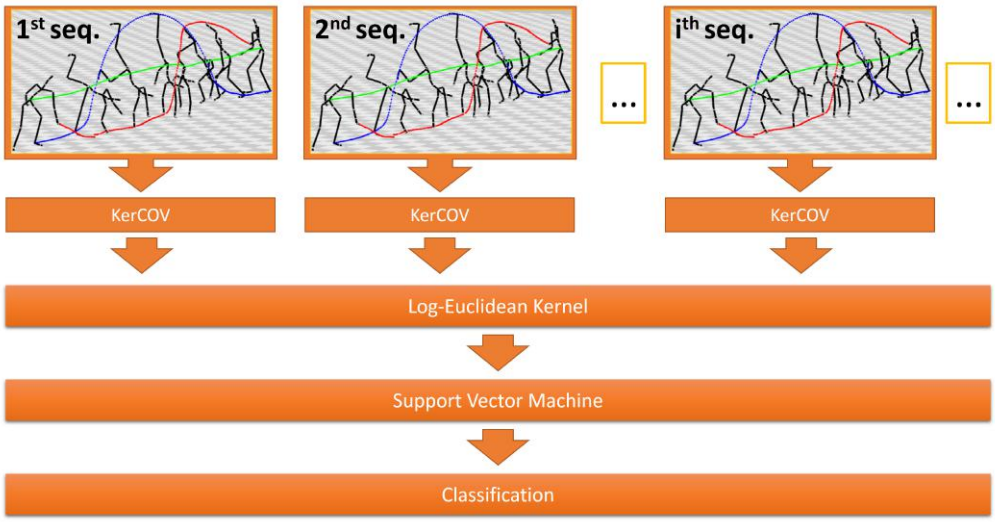


FIGURE 3.4: For each action, we compute a kernelized covariance representation as in Algorithm (1). Then, we use a Log-Euclidean kernel to obtain a Gram matrix which is subsequently adopted to train a (non-linear) support vector machine for the final classification.

maps, from which 20 joints are extracted to model the human pose of any of the human agents.

- MSR-Daily-Activity [Wan+12d], captured by using a Kinect device and it is composed by 16 different classes related to every-day actions such as *read book* or *lie down on sofa*. All of them are performed by 10 subjects. The main difficulty of this dataset originates from the fact that any activity class is performed in an either standing/sitting position, with a consequent misleading motion pattern to mess up the classification.
- MSRC-Kinect12 [Fot+12], consisting of sequences of human movements, represented as body-part locations, and the associated gesture to be recognized by the system. 594 sequences of approximate total length of six hours and 40 minutes are collected from 30 people performing 12 gestures: in total, 6,244 gesture instances. The motion files contain Kinect estimated trajectories of 20 joints.
- HDM-05 [M+07], containing more than tree hours of systematically recorded and well-documented MoCap data using a 240Hz VICON system to acquire the gestures of 5 non-professional actors via 31 markers. Motion clips have been manually cut out and annotated into roughly 100 different motion classes: on average, 10-50 realizations per class are available.

In all cases, we used the same splits adopted in [Wan+15b]: for MSR-Action3D, MSR-Daily-Activity and MSRC-Kinect12, training is performed on odd-index subject, while the even-index ones are left for testing (cross-subject pipeline of [Li+10b]), while, in HDM-05, the training split exploits all the data from the "bd" and "mm" subjects and testing is performed on "bk", "dg" and "tr".

Furthermore, for the HDM-05 dataset we removed some severely corrupted samples [Hus+13] and, as performed by [Wan+15b], selected only the following classes: *clap above head*, *deposit floor*, *elbow to knee*, *grab high*, *hop both legs*,

jog, *kick forward*, *lie down floor*, *rotate both arms backward*, *sit down chair*, *sneak*, *squat*, *stand up lie* and *throw basketball*. All the data are pre-processed in a common way. In particular, in MSR-Action3D and MSR-Daily-Activity, we computed the velocity and acceleration from the raw positions of the joints adopting either first and second order finite different scheme respectively as in [Zan+13].

Table 3.1 shows the results of *Kernelized-COV* on the four different datasets in comparison with all the other methods. Therein, in the case of MSR-Action3D and MSR-Daily-Activity, our proposed method is able to achieve comparable results with a small deviation from the state-of-the-art [Wan+15b], but it outperforms all the other competitors. More impressively, on MSRC-Kinect12, *Kernelized-COV* improves the-state-of-the-art [Wan+15b] by 2.7%. Even in the last dataset, namely HDM-05, the accuracy of the proposed method is 1.3% higher of the best score achieved by the other competitors. In this case, referring to [Hus+13], we did not report the accuracy on HDM-05 due to the different experimental settings: *Hierarchy of COVs* scored 95.41% on a simplified 11-class problem, while, in the same conditions, we scored 98.8%. Furthermore, it is worth noting that, on all the considered datasets our *Kernelized-COV* works even better than a recent infinite covariance operator [Har+14], more discriminatively encoding the data.

The improvements in classification accuracies demonstrate the effectiveness of *Kernelized-COV*. Moreover, our proposed principled way of encoding nonlinearities conveyed by the data is always superior to classical covariance based methods such as [Tuz+06b; Hus+13; Har+14] and does not suffer the gap in performance showed by covariance representation in [Wan+15b].

Method	MSR-Action3D
Action Graph [Li+10a]	79.0%
Random Occupancy Patterns [Wan+12c]	86.0%
Actionlets [Wan+12b]	88.2%
Pose Set [Wan+13b]	90.0%
Moving Pose [Zan+13]	91.7%
Lie Group [Vem+14]	92.5%
Normal Vectors [YT14b]	93.1%
Kernelized-COV (proposed)	**96.2%**

TABLE 3.2: Comparison against other classical approaches for action and activity recognition from MoCap data.

As a final remark, it is interesting to compare the performance of our *Kernelized-COV* with other not covariance-based methods. To this aim, we take into account the MSR-Action3D dataset and we compared with many previous approaches in the literature. From this analysis, the results presented in Table 3.2 give a further evidence of the effectiveness of the proposed use of the kernelized covariance, which is able to overcome [YT14b], the best score reported, by a margin of 3.1%.

3.1.2 Conclusions

We present a principled mathematical paradigm to recover the applicability of kernel trick for covariance matrix, in order to better model more general class of relationships other than the linear ones. This enhances the descriptiveness of the classical covariance matrix which is retrievable as a particular case of our general theoretical framework. Experimentally, *Kernelized-COV* closes the gap between covariance and kernel-based representations in many action recognition datasets, namely MSR-Action3D, MSR-Daily-Activity, MSRC-Kinect12 and HDM-05. The proposed method is able to improve the previous best accuracies, setting the new state-of-the-art performance on the last two datasets.

As a future work, we either tackle the applicability of this novel framework to other classification problems and we will also investigate how a similar pipeline can be extended to more general classes of kernel functions.

3.2 Approximated kernel machines for scalable and compact covariance-based temporal representations

Action recognition is a paramount research domain in machine intelligence and computer vision, being nowadays ubiquitous in human-robot interaction, autonomous driving, elderly care and video-surveillance, just to name a few applicative domains [Moe+06]. Yet, major difficulties arise when dealing with videos due to general visual ambiguities such as illumination variations, the presence of clutter/noise in the scene, occlusions or unfavorable recording viewpoint. Moreover, the variability of action evolution, as either executed by different human subjects or implicit in the structure of the action execution, further contributes to complicate the classification process. Fortunately, the adoption of novel range sensors constitutes an effective countermeasure as they provide alternative data to process, more robust to the above mentioned issues. Actually, with these sensors, a given action can be represented as a collection of skeletal joint positions progressing in time. Action recognition can thus be reformulated as the problem of classifying the multivariate time-series $\mathbf{P} \in \mathbb{R}^{3J \times T}$, which collect the three-dimensional coordinates of the J skeletal joints positions over T temporal acquisitions.

Within the data structure $\mathbf{P}$, J is fixed by the selection of the device which acquires the joints (*e.g.* , Kinect or VICON), while T typically changes across instances. Therefore, a minimal requirement for encoding this data is to be invariant to the variability of T. Among the possible feature encoding methods (see [Moe+06] for a literature review), the symmetric and positive definite (SPD) covariance (COV) operator guarantees this property, while also demonstrated to score a solid performance in 3D action recognition [Har+14; Cav+16; Cav+17a; Cav+17c]. In fact, in addition to properly modeling the skeletal dynamics with a second order statistics, the COV operator is also naturally able to handle different temporal durations of the action instances. This avoids slow pre-processing stages such as time warping or interpolation [Vem+14], needed to "re-align" the different sequences before actual classification. Moreover, performance achieved by COV-based methods are always comparable and sometimes superior to the one achieved by deep learning methods [Sha+16; Liu+16;

Liu+17a; Ke+17; HG17b; Hua+17; Wan+16; Li+17a], which, instead, typically require a massive amount of data and large computational power (on GPUs) for training.

All covariance-based paradigms for action recognition can be framed as the problem of classifying $d \times d$ data instances $\mathbf{X}$. In the case of skeleton data, $d = 3J$ and $\mathbf{X} = \frac{1}{T-1}\mathbf{P}\mathbf{J}\mathbf{P}^\top$, where $\mathbf{J} = \frac{1}{T}\mathbf{I} - 1_{T\times T}$ (being $\mathbf{I}$ the identity matrix) is the centering matrix as defined in [Min+14b; Min+16b]. To accomplish such task, kernel theory [Lea] naturally promotes max-margin approaches, in order to learn decision boundaries which maximally separate (action) classes. Interestingly, this can be done by *only* evaluating a kernel function K that, in this Section, we will be fixed as the Radial Basis Function (RBF) Gaussian kernel:

$$K(\mathbf{X}, \mathbf{Y}) = \exp\left(-\frac{1}{2\sigma^2}\|\mathbf{X} - \mathbf{Y}\|_F^2\right). \tag{3.18}$$

The choice of this kernel function is motivated by a set of beneficial properties, *i.e.* , 1) invariance to translations, 2) isotropy and 3) infinite-smoothness. Moreover, due to its robustness with respect to the parameter σ, it has been broadly and effectively used in the literature for many tasks [Lea; RR07; KK12; Vem+10; VZ12; Min+14b; Rin; Min+16b]. As a second motivation, after the change of variables $\mathbf{X} = \log(\frac{1}{T-1}\mathbf{P}\mathbf{J}\mathbf{P}^\top)$, equation (3.18) becomes the log-Euclidean kernel, which, thanks to its strong theoretical properties, is well suited to compare SPD matrices [Ars+07]. To this end, it has been widely exploited in computer vision and related fields, such as action recognition [Cav+16] or pedestrian re-Identification [Tos+13], to name a few.

Unfortunately, this approach has a limited scalability, since (3.18) has to be computed for each pair of examples within the training set $\{\mathbf{X}_1, \ldots, \mathbf{X}_N\}$ and for each ordered pair across training and test sets $\{\mathbf{Y}_1, \ldots, \mathbf{Y}_M\}$. This yields to the training and test Gram matrices

$$\begin{bmatrix} K(\mathbf{X}_1, \mathbf{X}_1) & K(\mathbf{X}_1, \mathbf{X}_2) & \ldots & K(\mathbf{X}_1, \mathbf{X}_N) \\ K(\mathbf{X}_2, \mathbf{X}_1) & K(\mathbf{X}_2, \mathbf{X}_2) & \ldots & K(\mathbf{X}_2, \mathbf{X}_N) \\ \vdots & \vdots & \ddots & \vdots \\ K(\mathbf{X}_N, \mathbf{X}_1) & K(\mathbf{X}_N, \mathbf{X}_2) & \ldots & K(\mathbf{X}_N, \mathbf{X}_N) \end{bmatrix} \tag{3.19}$$

and

$$\begin{bmatrix} K(\mathbf{Y}_1, \mathbf{X}_1) & K(\mathbf{Y}_1, \mathbf{X}_2) & \ldots & K(\mathbf{Y}_1, \mathbf{X}_N) \\ K(\mathbf{Y}_2, \mathbf{X}_1) & K(\mathbf{Y}_2, \mathbf{X}_2) & \ldots & K(\mathbf{Y}_2, \mathbf{X}_N) \\ \vdots & \vdots & \ddots & \vdots \\ K(\mathbf{Y}_M, \mathbf{X}_1) & K(\mathbf{Y}_M, \mathbf{X}_2) & \ldots & K(\mathbf{Y}_M, \mathbf{X}_N). \end{bmatrix} \tag{3.20}$$

In the case of large number of samples M and/or N, Gram matrices are quite hard to both store and manipulate when performing the optimization to determine the decision boundaries. For instance, if $M, N \sim 10^4$, about 10^{12} products are required to perform a matrix inversion, which will likely result in an out-of-memory error.

Such problem can be circumvented if we are able to obtain an explicitly computable feature representation $\boldsymbol{\phi}$ such that $\langle \boldsymbol{\phi}(\mathbf{X}), \boldsymbol{\phi}(\mathbf{Y}) \rangle$ equals (3.18), even approximately. In fact, while a linear machine fed with $\mathbf{X}$ is theoretically equivalent to a kernel machine (thanks to the kernel trick [Lea]), training a linear SVM is scalable even in the big data regime, differently from an exact

kernel SVM [Lee+15; Chi+16]. However, despite a few approximation schemes have been proposed [RR07; KK12; Vem+10; VZ12; Le+13], there is not yet a definitive answer about which compact and scalable classification pipeline performs the best in applicative settings.

In this Section, we will tackle all previous issues through the following main contributions.

1. We propose a novel, explicit *random* feature map, which can rigorously be interpreted as a compact approximation inspired by the exact (and infinite-dimensional) feature encoding induced by (3.18).
2. We theoretically show that, marginalizing the sources of randomness, the proposed estimator of (3.18) is unbiased, and its variance has an explicit upper bound that is i) more clearly interpretable and ii) more rapidly decreasing as a function of the size of the approximation. These properties make our approach more favorable with respect to competing methods in the literature [RR07; KK12; Vem+10; VZ12; Le+13].
3. We present an extensive experimental comparison between existing approximation schemes on common benchmark datasets on which our method assesses its superiority.
4. In a broad experimental validation on a consistent number of publicly available benchmark datasets, we demonstrate the superiority of our proposed approach for 3D action recognition, in terms of ease and speed of training, compactness of the representation and improvement over state-of-the-art performance.

3.2.1 Approximating the RBF kernel with Kronecker products

In this Section, we present in formal terms our original technique to approximate the RBF kernel (3.18) by means of a low-dimensional and explicit feature map, characterized by a random component which is ultimately responsible of the quality of the approximation itself. Indeed, when averaging upon all the possible realization of such component, our representation approximates (3.18) with zero bias. Additionally, the variance of such estimation can be controlled by an explicit upper bound that easily writes as a function which rapidly decreases as the feature dimensionality increases.

Construction of the approximated feature map.

Given $\mathbf{X} \in \mathbb{R}^{d \times d}$ and fixed a strictly positive integer ν, that corresponds to the feature dimensionality, our approximation is defined as follows.

Definition 1. *We define a ν dimensional vector $\boldsymbol{\phi}_{\text{kron}-\pi}(\mathbf{X})$ whose components $\phi_{\text{kron}-\pi,1}(\mathbf{X}), \dots, \phi_{\text{kron}-\pi,\nu}(\mathbf{X})$ are ($1/\sqrt{\nu}$-multiplied) independent realizations of the following scalar function*

$$\varphi_{\text{kron}-\pi}(\mathbf{X}) = \frac{1}{\sigma^{2n}} \sqrt{\frac{\exp(-\frac{1}{\sigma^2})}{\rho(n) n!}} \operatorname{tr}\left(\otimes_{\kappa=1}^{n} \mathbf{W}^{(\kappa)\top} \mathbf{X} \right). \tag{3.21}$$

In (3.21), $\sigma > 0$ *defines the bandwidth of the kernel function* (3.18), n *is sampled from* any *distribution* ρ *supported over the integers. Furthermore, the following assumptions are made:*

A.1 $\mathbf{W}^{(\kappa)}$ *are (elementwise) drawn from the distribution* $\mathcal{P}$ *with null expected value and standard deviation equals to the kernel's bandwidth* σ.

A.2 The $d \times d$ *matrix which is inputted to* $\boldsymbol{\phi}_{\text{kron}-\pi}$ *lies on the Frobenius norm-unitary sphere, that is* $\|\mathbf{X}\|_F = 1$.

Note that the $\boldsymbol{\phi}_{\text{kron}-\pi}(\mathbf{X})$ has two sources of randomness. First, the integer n, which is sampled from ρ. Second, precisely n matrices $\mathbf{W}^{(1)}, \ldots, \mathbf{W}^{(\kappa)}, \ldots, \mathbf{W}^{(n)}$ are sampled, so that each of their element is independently drawn from $\mathcal{P}$. More in detail, for each $\kappa = 1, \ldots, n$, the transpose of $\mathbf{W}^{(\kappa)}$ is (row-by-column) multiplied by $\mathbf{X}$. Afterwards, the results of the previous operation are combined together with a Kronecker product and, finally, the trace operator is evaluated. For the sake of clarity, let us notice that, since the trace operator applied on matrix returns a scalar, $\boldsymbol{\phi}_{\text{kron}-\pi}(\mathbf{X}) \in \mathbb{R}$ and $\boldsymbol{\phi}_{\text{kron}-\pi}(\mathbf{X}) \in \mathbb{R}^v$, since it stacks v independent realizations of $\boldsymbol{\phi}_{\text{kron}-\pi}(\mathbf{X})$ (divided by $\sqrt{v}$, which is factorized out of the definition of φ only for convenience in the demonstrations). Algorithm 2 provides the pseudo-code for the construction process.

With respect to the assumptions A.1 and A.2, the first one constrains the distribution $\mathcal{P}$. Indeed, let us notice that, in all our theoretical exposition, the distributions ρ and $\mathbf{P}$ are allowed to be highly general, and we will specify them only in the experiments when we need to numerically sample from them. For instance, A.1 is satisfied if $\mathcal{P} = \mathcal{N}(0, \sigma^2)$, being fixed as a zero-mean Gaussian with σ^2 variance.

Instead, A.2 is only technical and does not really represent a constraint under an applicative point of view. Indeed, given an arbitrary input data $\mathbf{X}$, we can achieve A.2 by dividing $\mathbf{X}$ entrywise by $\|\mathbf{X}\|_F$. Such operation is easy to perform and it is along the line of the classical pre-processing which is applied on the data before passing them to a kernel method - as for instance, the component-wise division by the standard deviation is a common preprocessing step before SVM training [Lea]. If compared with similar results in [RR07; Vem+10; VZ12; KK12; Le+13], the assumption of unitary norm for $\mathbf{X}$ and $\mathbf{Y}$ is in line with the analogous assumptions of sampling the data from a given submanifold - with the remarkable difference that our assumption is easy to satisfy also in an applicative domain.

Before digging into the details of the theoretical foundation, lets us provide the intuition behind equation (3.21).

Intuition behind the genesis of $\varphi_{\text{kron}-\pi}$

According to the well established kernel theory [Lea], the exact feature map $\mathbf{f}$ associated to the RBF kernel (3.18) is infinite-dimensional. Still, it can be expressed in closed form. In fact, without loss of generality, let us assume $d = 1$ and, for the sake of simplicity, let $\sigma = 1$. Consequently, we replace the matrices $\mathbf{X}, \mathbf{Y}$ with the scalars x, y and, in such a case, the kernel function (3.18) rewrites as $K(x, y) = \exp(-\frac{1}{2}(x - y)^2)$.

Algorithm 2: Approx, by Kronecker product.

Input: A normalized $d \times d$ input matrix $\mathbf{X}$, the desired feature size ν, the probability distributions ρ over integers and $\mathcal{P}$ over real numbers, the kernel bandwith $\sigma > 0$.

Output: $[\phi_{\text{kron}-\pi,1}(\mathbf{X}), \dots, \phi_{\text{kron}-\pi,\nu}(\mathbf{X})]$

foreach $j = 1, \dots, \nu$ **do**

  Sample n according to ρ

  foreach $\kappa = 1, \dots, n$ **do**

    Sample $\mathbf{W}^{(\kappa)} \in \mathbb{R}^{d \times d}$ from $\mathcal{P}$ elementwise.

  end

  Compute the scalar $\pi(\mathbf{X}) = \text{tr}\left(\otimes_{\kappa=1}^{n} {\mathbf{W}^{(\kappa)}}^{\top} \mathbf{X}\right)$

  Return $\phi_{\text{kron}-\pi,j}(\mathbf{X}) = \sigma^{-2n} \left(\frac{\exp(-\sigma^{-2})}{\nu \rho(n) n!}\right)^{1/2} \pi(\mathbf{X})$

end

We would like to write the exact infinite dimensional feature map $x \mapsto \mathbf{f}(x)$ for such RBF kernel, i.e. the exact infinite-dimensional vector $\mathbf{f}(\cdot)$ such that

$$\langle \mathbf{f}(x), \mathbf{f}(y) \rangle = K(x, y) = \exp(-(x-y)^2/2) \tag{3.22}$$

where the inner product $\langle \cdot, \cdot \rangle$ is computed over the square-integrable series of $\mathbf{f}(\cdot)$. Since

$$\exp(-(x-y)^2/2) = \exp(-x^2/2) \cdot \exp(xy) \cdot \exp(-y^2/2), \tag{3.23}$$

we can take advantage of the Taylor expansion to obtain

$$\mathbf{f}(x) = \sqrt{e^{-x^2}} \left[1, x, \frac{x^2}{\sqrt{2!}}, \frac{x^3}{\sqrt{3!}}, \dots, \frac{x^n}{\sqrt{n!}}, \dots\right]. \tag{3.24}$$

As certified by Lagrange's remainder formula for Taylor expansions [Rud66], a good approximation of (3.24) is obtained by considering all the terms which are less or equal to a certain degree n. In the scalar case, those terms are exactly n. Differently, in order to compute the products when $d > 1$, the terms of a given degree n must include all the possible combinations $X_{11}^{\alpha_{11}} X_{12}^{\alpha_{12}} \cdots X_{ij}^{\alpha_{ij}} \cdots X_{dd}^{\alpha_{dd}}$, where X_{ij} are the components of $\mathbf{X}$ and α_{ij} are d^2 non-negative integers such that $\sum_{ij} \alpha_{ij} = n$. That is, we have to consider all the $n / \prod_{ij} \alpha_{ij}!$ combinations, and this has an exponential complexity with respect to d [Lea]. This clearly produces an exponentially-sized feature map that, as shown in [Rin], is formally fine but obviously not applicable in real-world datasets. In fact, as the operative condition assumed in [Rin], d needs to be less than 4.

Since the analytical pipeline inspired by Taylor's remainder theorem is not viable in practical pattern analysis, in this Chapter we propose a manageable (alternative) solution. When asked to build a ν-dimensional representation, we repeat ν times the following pipeline. We sample n from ρ and we use n as a pointer to index which component of (3.24) to sample. Then, as a surrogate technique for computing all the possible combinations of products of degree n, we introduce $\otimes_{\kappa=1}^{n} {\mathbf{W}^{(\kappa)}}^{\top} \mathbf{X} = \mathbf{W}^{(1)\top} \otimes \mathbf{X} \otimes \dots \mathbf{W}^{(n)\top} \otimes \mathbf{X}$. The latter is directly inspired from the technique of random rescaling, which is common practice in

random approximated feature map approaches [RR07; Vem+10; VZ12; Le+13; KK12], where introducing random projections can be interpreted as a trick to "recover" from the sparse sampling of n. In the limit case where $\mathbf{W}^{(\kappa)}$ are identity matrices, $\otimes_{\kappa=1}^{n} \mathbf{W}^{(\kappa)\top} \mathbf{X} = \mathbf{X}^{\otimes n}$ and we can find a clear analogy between scalar exponentiation in (3.24) and Kronecker exponentiation in (3.21), being the latter a $d \times d$ generalization of the former.

Formal proofs: unbiasedness and variance bound

In this Section, we demonstrate that, thanks to assumptions A.1 and A.2, once averaging upon all possible realizations of n from ρ and $\mathbf{W}^{(1)}, \dots, \mathbf{W}^{(n)}$ from $\mathcal{P}$, we have no bias in approximating the kernel - that is, the expected value of our $\langle \boldsymbol{\phi}_{\text{kron}-\pi}(\mathbf{X}), \boldsymbol{\phi}_{\text{kron}-\pi}(\mathbf{Y}) \rangle$ coincides with (3.18) and, at the same time, we are able to control the variance of the estimation.

Unbiasedness of $\boldsymbol{\phi}_{\text{kron}-\pi}$. As previously explained, an exact feature map $\mathbf{f}$ is able to satisfy the equality $\langle \mathbf{f}(\mathbf{X}), \mathbf{f}(\mathbf{Y}) \rangle = K(\mathbf{X}, \mathbf{Y})$. Thanks to the well established kernel trick [Lea], one does not need to compute $\mathbf{f}$ explicitly but, instead, a kernel machine can be trained by evaluating the kernel function only. In many cases (like the one of RBF kernel (3.18)), computing $\mathbf{f}$ explicitly is impossible due to its infinite dimension. Moreover, on the opposite, computing the kernel function does not scale to big datasets, since evaluating $K(\mathbf{X}, \mathbf{Y})$ for *every* $\mathbf{X}$ and $\mathbf{Y}$ has a quadratic complexity. Due to the prohibitive size of the Gram matrices (3.19) and (3.20), either the training or inference stages may be simply not computationally affordable (typically because of out-of-memory issues).

In order to accommodate for that, we propose to replace $\mathbf{f}$ with a map $\boldsymbol{\phi}_{\text{kron}-\pi}$, such that

$$\langle \boldsymbol{\phi}_{\text{kron}-\pi}(\mathbf{X}), \boldsymbol{\phi}_{\text{kron}-\pi}(\mathbf{Y}) \rangle \approx K(\mathbf{X}, \mathbf{Y}) \tag{3.25}$$

with the crucial difference that $\boldsymbol{\phi}$ is explicitly computable. In other words, while the kernel trick allows to replace the feature map $\mathbf{f}$ with the kernel function K, we revert the perspective, and evaluate the kernel function with $\boldsymbol{\phi}_{\text{kron}-\pi}$, which, differently from $\mathbf{f}(\mathbf{X})$, is finite-dimensional and explicitly computable. In fact, a linear model fed with $\boldsymbol{\phi}_{\text{kron}-\pi}$ is a theoretically valid estimate for the exact kernel machine fed with (3.18).

As well established in the literature that similarly proposed random approximated feature maps [RR07; KK12; Le+13; Vem+10; VZ12], we want to demonstrate the validity of the approximation by showing that, once averaging upon all the sources of randomness which affect our feature map $\boldsymbol{\phi}_{\text{kron}-\pi}$, an equality holds in eq. 3.25. In other words, we want to prove the absence of biases in the approximation.

Theorem 2 (Unbiased approximation for $\boldsymbol{\phi}_{\text{kron}-\pi}$). *With the previous notations, the linear kernel $\langle \boldsymbol{\phi}_{\text{kron}-\pi}(\mathbf{X}), \boldsymbol{\phi}_{\text{kron}-\pi}(\mathbf{Y}) \rangle$ induced by $\boldsymbol{\phi}_{\text{kron}-\pi}$ is an unbiased estimator for $K(\mathbf{X}, \mathbf{Y})$ as in (3.18). Indeed,*

$$\mathbb{E}_{n,\mathcal{P}} \left[\langle \boldsymbol{\phi}_{\text{kron}-\pi}(\mathbf{X}), \boldsymbol{\phi}_{\text{kron}-\pi}(\mathbf{Y}) \rangle \right] = K(\mathbf{X}, \mathbf{Y}), \tag{3.26}$$

being the expected value jointly computed over all possible realizations of n from ρ and of $\mathbf{W}^{(\kappa)}$ from $\mathcal{P}$, $\kappa = 1, \dots, n$.

Proof. Fix two arbitrary $d \times d$ matrices $\mathbf{X}$ and $\mathbf{Y}$. Let $\boldsymbol{\phi}_{\text{kron}-\pi}(\mathbf{X})$ and $\boldsymbol{\phi}_{\text{kron}-\pi}(\mathbf{Y})$ computed according to Algorithm 2. We are interested in inspecting the linear kernel $\langle \boldsymbol{\phi}_{\text{kron}-\pi}(\mathbf{X}), \boldsymbol{\phi}_{\text{kron}-\pi}(\mathbf{Y}) \rangle$, induced by $\boldsymbol{\phi}_{\text{kron}-\pi}$, as to compute its expected value while averaging upon all the possible realizations of $n \sim \rho$ and $\mathbf{W}^{(\kappa)^\top} \sim \mathcal{P}$ element-wise, $\kappa = 1, \dots, n$. Due to the linearity of expectation,

$$\mathbb{E}_{n,\mathcal{P}} \left[\langle \boldsymbol{\phi}_{\text{kron}-\pi}(\mathbf{X}), \boldsymbol{\phi}_{\text{kron}-\pi}(\mathbf{Y}) \rangle \right] = \sum_{\ell=1}^{\nu} \mathbb{E}_{n,\mathcal{P}} \left[\phi_{\text{kron}-\pi,\ell}(\mathbf{X}) \cdot \phi_{\text{kron}-\pi,\ell}(\mathbf{Y}) \right] \quad (3.27)$$

and, also,

$$\mathbb{E}_{n,\mathcal{P}} \left[\langle \boldsymbol{\phi}_{\text{kron}-\pi}(\mathbf{X}), \boldsymbol{\phi}_{\text{kron}-\pi}(\mathbf{Y}) \rangle \right] = \nu \mathbb{E}_{n,\mathcal{P}} \left[\varphi_{\text{kron}-\pi}(\mathbf{X})/\sqrt{\nu} \cdot \varphi_{\text{kron}-\pi}(\mathbf{Y})/\sqrt{\nu} \right], \quad (3.28)$$

since, by construction, each component of $\boldsymbol{\phi}_{\text{kron}-\pi}$ is an independent identical realization of $\varphi_{\text{kron}-\pi}$, divided by $\sqrt{\nu}$.

By simplifying the factor ν - which is possible due to the linearity of the expected value - and by exploiting the definition of $\varphi_{\text{kron}-\pi}$, we can expand the expected value as follows

$$\begin{aligned} &\mathbb{E}_{n,\mathcal{P}} \left[\langle \boldsymbol{\phi}_{\text{kron}-\pi}(\mathbf{X}), \boldsymbol{\phi}_{\text{kron}-\pi}(\mathbf{Y}) \rangle \right] = \\ &\quad = \exp(-\tfrac{1}{\sigma^2}) \sum_{n=0}^{\infty} \frac{1}{\sigma^{4n}} \frac{1}{n!} \mathbb{E}_{\mathcal{P}} \left[\operatorname{tr} \left(\otimes_{\kappa=1}^{n} \mathbf{W}^{(\kappa)^\top} \mathbf{X} \right) \operatorname{tr} \left(\otimes_{\kappa=1}^{n} \mathbf{W}^{(\kappa)^\top} \mathbf{Y} \right) \right], \end{aligned}$$

where the weights $\mathbf{W}^{(\kappa)^\top}$ are shared since we are evaluating one realization of $\varphi_{\text{kron}-\pi}$ and where we simplified the distribution $1/\rho(n)$ with the factor $\rho(n)$ which arises when we compute the expected value over ρ.

Now, exploit the property $\operatorname{tr}(\mathbf{A} \otimes \mathbf{B}) = \operatorname{tr}(\mathbf{A}) \operatorname{tr}(\mathbf{B})$. Then,

$$\begin{aligned} &\mathbb{E}_{n,\mathcal{P}} \left[\langle \boldsymbol{\phi}_{\text{kron}-\pi}(\mathbf{X}), \boldsymbol{\phi}_{\text{kron}-\pi}(\mathbf{Y}) \rangle \right] = \\ &\quad = \exp(-\tfrac{1}{\sigma^2}) \sum_{n=0}^{\infty} \frac{1}{\sigma^{4n}} \frac{1}{n!} \mathbb{E}_{\mathcal{P}} \left[\prod_{\kappa=1}^{n} \operatorname{tr} \left(\mathbf{W}^{(\kappa)^\top} \mathbf{X} \right) \prod_{\kappa=1}^{n} \operatorname{tr} \left(\mathbf{W}^{(\kappa)^\top} \mathbf{Y} \right) \right] \end{aligned}$$

and, by exploiting the independence of the weights $\mathbf{W}^{(\kappa)^\top}$ - for a fixed κ, the properties of the expected value allow permute products and expectations, achieving

$$\begin{aligned} &\mathbb{E}_{n,\mathcal{P}} \left[\langle \boldsymbol{\phi}_{\text{kron}-\pi}(\mathbf{X}), \boldsymbol{\phi}_{\text{kron}-\pi}(\mathbf{Y}) \rangle \right] = \\ &\quad = \exp(-\tfrac{1}{\sigma^2}) \sum_{n=0}^{\infty} \frac{1}{\sigma^{4n}} \frac{1}{n!} \prod_{\kappa=1}^{n} \mathbb{E}_{\mathcal{P}} \left[\operatorname{tr} \left(\mathbf{W}^{(\kappa)^\top} \mathbf{X} \right) \cdot \operatorname{tr} \left(\mathbf{W}^{(\kappa)^\top} \mathbf{Y} \right) \right]. \end{aligned}$$

Using the property that $\operatorname{tr}(\mathbf{AB}) = \langle \mathbf{A}, \mathbf{B} \rangle_F$, using the explicit expression for the Frobenius inner product, we achieve

$$\begin{aligned} &\mathbb{E}_{n,\mathcal{P}} \left[\langle \boldsymbol{\phi}_{\text{kron}-\pi}(\mathbf{X}), \boldsymbol{\phi}_{\text{kron}-\pi}(\mathbf{Y}) \rangle \right] = \\ &\quad = \exp(-\tfrac{1}{\sigma^2}) \sum_{n=0}^{\infty} \frac{1}{\sigma^{4n}} \frac{1}{n!} \prod_{\kappa=1}^{n} \mathbb{E}_{\mathcal{P}} \left[\sum_{i,j,h,k=1}^{d} W_{ij}^{(\kappa)} X_{ij} W_{hk}^{(\kappa)} Y_{hk} \right] \end{aligned}$$

and

$$\mathbb{E}_{n,\mathcal{P}}\left[\langle \boldsymbol{\phi}_{\text{kron}-\pi}(\mathbf{X}), \boldsymbol{\phi}_{\text{kron}-\pi}(\mathbf{Y})\rangle\right] = \\ = \exp(-\tfrac{1}{\sigma^2}) \sum_{n=0}^{\infty} \frac{1}{\sigma^{4n}} \frac{1}{n!} \prod_{\kappa=1}^{n} \sum_{i,j,h,k=1}^{d} \mathbb{E}_{\mathcal{P}}\left[W_{ij}^{(\kappa)} W_{hk}^{(\kappa)}\right] X_{ij} Y_{hk} \tag{3.29}$$

by linearity of expectation. Since, by assumption, $W_{ij}^{(\kappa)}$ are independently and identically sampled from the zero-mean distribution $\mathcal{P}$, we get

$$\mathbb{E}_{\mathcal{P}}\left[W_{ij}^{(\kappa)} W_{hk}^{(\kappa)}\right] = \text{var}[\mathcal{P}]\delta_{ij}\delta_{hk}, \tag{3.30}$$

where, since all $W_{ij}^{(\kappa)}$ are identically distributed, we can generically replace the variance of each of those with the variance $\text{var}[\mathcal{P}]$ of the scalar distribution $\mathcal{P}$ from which $W_{ij}^{(\kappa)}$ are all independently sampled. Also, for the sake of clarity, let us recall that, in (3.30), δ_{ij} denotes the Kronecker symbol, being $\delta_{ij} = 1$ if $i = j$ and 0 otherwise.

Therefore, once plugged (3.30) into (3.29), we can use Kronecker's symbol to achieve

$$\mathbb{E}_{n,\mathcal{P}}\left[\langle \boldsymbol{\phi}_{\text{kron}-\pi}(\mathbf{X}), \boldsymbol{\phi}_{\text{kron}-\pi}(\mathbf{Y})\rangle\right] = \exp(-\tfrac{1}{\sigma^2}) \sum_{n=0}^{\infty} \frac{1}{\sigma^{4n}} \frac{1}{n!} \prod_{\kappa=1}^{n} \text{var}[\mathcal{P}]^n \sum_{i,j=1}^{d} X_{ij} Y_{ij}, \tag{3.31}$$

that is

$$\mathbb{E}_{n,\mathcal{P}}\left[\langle \boldsymbol{\phi}_{\text{kron}-\pi}(\mathbf{X}), \boldsymbol{\phi}_{\text{kron}-\pi}(\mathbf{Y})\rangle\right] = \text{var}[\mathcal{P}] \frac{\exp(-\frac{1}{\sigma^2})}{\sigma^{4n}} \sum_{n=0}^{\infty} \frac{1}{n!} \left(\langle \mathbf{X}, \mathbf{Y}\rangle_F\right)^n. \tag{3.32}$$

Apply the Taylor expansion for the exponential function, as well as the propery $e^a \cdot e^b = e^{a+b}$.

$$\mathbb{E}_{n,\mathcal{P}}\left[\langle \boldsymbol{\phi}_{\text{kron}-\pi}(\mathbf{X}), \boldsymbol{\phi}_{\text{kron}-\pi}(\mathbf{Y})\rangle\right] = \exp(-\tfrac{1}{\sigma^2}) \exp\left(\frac{\text{var}[\mathcal{P}]}{\sigma^4} \langle \mathbf{X}, \mathbf{Y}\rangle_F\right) \\ = \exp\left(\frac{\text{var}[\mathcal{P}]\langle \mathbf{X}, \mathbf{Y}\rangle_F - \sigma^2}{\sigma^4}\right).$$

Apply A.1, simplify a factor $\sigma^2 \neq 0$ and multiply and divide the fraction inside the exp function by 2.

$$\mathbb{E}_{n,\mathcal{P}}\left[\langle \boldsymbol{\phi}_{\text{kron}-\pi}(\mathbf{X}), \boldsymbol{\phi}_{\text{kron}-\pi}(\mathbf{Y})\rangle\right] = \exp\left(\frac{2\langle \mathbf{X}, \mathbf{Y}\rangle_F - 2}{2\sigma^2}\right). \tag{3.33}$$

Finally, thanks to A.2,

$$\mathbb{E}_{n,\mathcal{P}}\left[\langle \boldsymbol{\phi}_{\text{kron}-\pi}(\mathbf{X}), \boldsymbol{\phi}_{\text{kron}-\pi}(\mathbf{Y})\rangle\right] = \exp\left(\frac{2\langle \mathbf{X}, \mathbf{Y}\rangle_F - \|\mathbf{X}\|_F^2 - \|\mathbf{Y}\|_F^2}{2\sigma^2}\right) \\ = \exp\left(\frac{1}{2\sigma^2} - \|\mathbf{X} - \mathbf{Y}\|_F^2\right) = K(\mathbf{X}, \mathbf{Y}),$$

which gives the thesis thanks to the generality of $\mathbf{X}$ and $\mathbf{Y}$. □

Bound on the variance for $\boldsymbol{\phi}_{\text{kron}-\pi}$. Theorem 2 guarantees that, on average, $\langle \boldsymbol{\phi}_{\text{kron}-\pi}(\mathbf{X}), \boldsymbol{\phi}_{\text{kron}-\pi}(\mathbf{Y}) \rangle$ is a good approximation for $K(\mathbf{X}, \mathbf{Y})$, since there is no bias. This is a strong and necessary assumption to ensure that our statistical estimator is reliable, but it does not take into account the variance, i.e. the *quality* of the approximation. Namely, even an unbiased estimator can heavily deviate from its expected value if there are no theoretical guarantees for its variance. We can prove that our estimator well behaves also in this respect, since $\boldsymbol{\phi}_{\text{kron}-\pi}$ induces a linear kernel whose variance can be upper bounded as follows.

Theorem 3 (Bound on the variance of $\boldsymbol{\phi}_{\text{kron}-\pi}$). *With the previous notation, the linear kernel $\langle \boldsymbol{\phi}_{\text{kron}-\pi}(\mathbf{X}), \boldsymbol{\phi}_{\text{kron}-\pi}(\mathbf{Y}) \rangle$ induced by $\varphi_{\text{kron}-\pi}$ has a controlled variance which is bounded by a linear function of the feature dimensionality ν. Precisely,*

$$\text{var}\left[\langle \boldsymbol{\phi}_{\text{kron}-\pi}(\mathbf{X}), \boldsymbol{\phi}_{\text{kron}-\pi}(\mathbf{Y}) \rangle\right] \leq \frac{C_\rho}{\nu^3} \exp\left(\frac{9\, m_4(\mathcal{P}) - 2\sigma^4}{\sigma^8}\right) \tag{3.34}$$

where the variance is computed over all possible realizations of n from ρ and all possible manners of sampling $\mathbf{W}^{(\kappa)}$ from $\mathcal{P}$ for each κ.

C_ρ is defined as

$$C_\rho = \sum_{n=0}^{\infty} \frac{1}{\rho(n) \cdot n!} \tag{3.35}$$

and $m_4(\mathcal{P})$ denotes the forth order moment of $\mathcal{P}$.

Proof. Fix $\mathbf{X}$ and $\mathbf{Y}$ to be arbitrary $d \times d$ matrices. Then,

$$\text{var}_{n,\mathcal{P}}\left[\langle \boldsymbol{\phi}_{\text{kron}-\pi}(\mathbf{X}), \boldsymbol{\phi}_{\text{kron}-\pi}(\mathbf{Y}) \rangle\right] = \nu \sum_{\ell=1}^{\nu} \text{var}_{n,\mathcal{P}}[\phi_{\text{kron}-\pi,j}(\mathbf{X}) \cdot \phi_{\text{kron}-\pi,j}(\mathbf{Y})] \tag{3.36}$$

because the variance of the sum of independent variables equals the sum of the variances of the variables. Also,

$$\text{var}_{n,\mathcal{P}}\left[\langle \boldsymbol{\phi}_{\text{kron}-\pi}(\mathbf{X}), \boldsymbol{\phi}_{\text{kron}-\pi}(\mathbf{Y}) \rangle\right] \leq \nu \sum_{\ell=1}^{\nu} \mathbb{E}_{n,\mathcal{P}}[\phi_{\text{kron}-\pi,j}(\mathbf{X})^2 \cdot \phi_{\text{kron}-\pi,j}(\mathbf{Y})^2] \tag{3.37}$$

by definition $\text{var}(Z) = \mathbb{E}(Z^2) - \mathbb{E}(Z)^2$ of variance for a scalar random variable Z. But now, by construction $\phi_{\text{kron}-\pi}$ is and independent realization of $\varphi_{\text{kron}-\pi}$, therefore

$$\text{var}_{n,\mathcal{P}}\left[\langle \boldsymbol{\phi}_{\text{kron}-\pi}(\mathbf{X}), \boldsymbol{\phi}_{\text{kron}-\pi}(\mathbf{Y}) \rangle\right] \leq \frac{1}{\nu^3} \mathbb{E}_{n,\mathcal{P}}[\varphi_{\text{kron}-\pi}(\mathbf{X})^2 \cdot \varphi_{\text{kron}-\pi}(\mathbf{Y})^2] \tag{3.38}$$

Hence, by comparing (3.38) and (3.34), we will be able to conclude if we show that

$$\mathbb{E}_{n,\mathcal{P}}[\varphi_{\text{kron}-\pi}(\mathbf{X})^2 \cdot \varphi_{\text{kron}-\pi}(\mathbf{Y})^2] \leq C_\rho \exp\left(\frac{9\, m_4(\mathcal{P}) - 2\sigma^4}{\sigma^8}\right) \tag{3.39}$$

By using the definition of $\varphi_{\text{kron}-\pi}$, we rewrite $\varphi_{\text{kron}-\pi}(\mathbf{X})^2 \cdot \varphi_{\text{kron}-\pi}(\mathbf{Y})^2$ as

$$\frac{\exp(-\frac{2}{\sigma^2})}{\sigma^{8n}} \frac{1}{(\rho(n)n!)^2} \text{tr}\left(\otimes_{\kappa=1}^{n} \mathbf{W}^{(\kappa)\top}\mathbf{X}\right)^2 \text{tr}\left(\otimes_{\kappa=1}^{n} \mathbf{W}^{(\kappa)\top}\mathbf{Y}\right)^2$$

and, therefore,

$$\begin{aligned}\mathbb{E}_{n,\mathcal{P}}[\varphi_{\text{kron}-\pi}(\mathbf{X}) \cdot \varphi_{\text{kron}-\pi}(\mathbf{Y})] = \\ = \sum_{n=0}^{\infty} \frac{\exp(-\frac{2}{\sigma^2})}{\sigma^{8n}} \frac{1}{\rho(n)(n!)^2} \mathbb{E}_{\mathcal{P}}\left[\text{tr}\left(\otimes_{\kappa=1}^{n} \mathbf{W}^{(\kappa)\top}\mathbf{X}\right)^2 \text{tr}\left(\otimes_{\kappa=1}^{n} \mathbf{W}^{(\kappa)\top}\mathbf{Y}\right)^2\right]\end{aligned}$$

Now, using the property $\text{tr}(\mathbf{A} \otimes \mathbf{B}) = \text{tr}(\mathbf{A})\text{tr}(\mathbf{B})$, rearranging the products and commuting products and expected value due to independence,

$$\begin{aligned}\mathbb{E}_{n,\mathcal{P}}[\varphi_{\text{kron}-\pi}(\mathbf{X}) \cdot \varphi_{\text{kron}-\pi}(\mathbf{Y})] = \\ = \sum_{n=0}^{\infty} \frac{\exp(-\frac{2}{\sigma^2})}{\sigma^{8n}} \frac{1}{\rho(n)(n!)^2} \prod_{\kappa=1}^{n} \mathbb{E}_{\mathcal{P}}\left[\text{tr}\left(\mathbf{W}^{(\kappa)\top}\mathbf{X}\right)^2 \text{tr}\left(\mathbf{W}^{(\kappa)\top}\mathbf{Y}\right)^2\right].\end{aligned}$$

If we write down the trace as the sum of the diagonal components in an explicit manner, we obtain

$$\begin{aligned}\mathbb{E}_{n,\mathcal{P}}[\varphi_{\text{kron}-\pi}(\mathbf{X}) \cdot \varphi_{\text{kron}-\pi}(\mathbf{Y})] = \\ = \sum_{n=0}^{\infty} \frac{\exp(-\frac{2}{\sigma^2})}{\sigma^{8n}} \frac{1}{\rho(n)(n!)^2} \prod_{\kappa=1}^{n} \mathbb{E}_{\mathcal{P}}\left[\left(\sum_{i,j=1}^{d} W_{ij}^{(\kappa)} X_{ij}\right)^2 \cdot \left(\sum_{i,j=1}^{d} W_{hk}^{(\kappa)} Y_{hk}\right)^2\right]\end{aligned}$$

and

$$\begin{aligned}\mathbb{E}_{n,\mathcal{P}}[\varphi_{\text{kron}-\pi}(\mathbf{X}) \cdot \varphi_{\text{kron}-\pi}(\mathbf{Y})] = \sum_{n=0}^{\infty} \frac{\exp(-\frac{2}{\sigma^2})}{\sigma^{8n}} \frac{1}{\rho(n)(n!)^2} \cdot \\ \cdot \prod_{\kappa=1}^{n} \mathbb{E}_{\mathcal{P}}\left[\left(\sum_{i,j=1}^{d} \left(W_{ij}^{(\kappa)} X_{ij}\right)^2 + 2 \sum_{i>i,j>j=1}^{d} W_{ij}^{(\kappa)} W_{ij}^{(\kappa)} X_{ij} X_{ij}\right) \cdot \right. \\ \left. \cdot \left(\sum_{h,k=1}^{d} \left(W_{hk}^{(\kappa)} Y_{hk}\right)^2 + 2 \sum_{h>h,k>k=1}^{d} W_{hk}^{(\kappa)} W_{hk}^{(\kappa)} Y_{ij} Y_{hk}\right)^2\right].\end{aligned}$$

Let us study $\mathbb{E}_{\mathcal{P}}\left[\left(\sum_{i,j=1}^{d} \left(W_{ij}^{(\kappa)} X_{ij}\right)^2\right) \cdot \left(\sum_{h,k=1}^{d} \left(W_{hk}^{(\kappa)} Y_{hk}\right)^2\right)\right]$ separately. In fact, since the other three addends can be rearranged in the same manner, the only difference is the multiplicative scalar factor in front (1 or 2) that we will account for later. We have

$$\begin{aligned}\mathbb{E}_{\mathcal{P}}\left[\left(\sum_{i,j=1}^{d} \left(W_{ij}^{(\kappa)} X_{ij}\right)^2\right) \cdot \left(\sum_{h,k=1}^{d} \left(W_{hk}^{(\kappa)} Y_{hk}\right)^2\right)\right] = \\ = \sum_{i,j,h,k=1}^{d} \mathbb{E}_{\mathcal{P}}\left[\left(\left(W_{ij}^{(\kappa)} X_{ij}\right)^2\right) \cdot \left(\left(W_{hk}^{(\kappa)} Y_{hk}\right)^2\right)\right],\end{aligned}$$

$$\mathbb{E}_{\mathcal{P}}\left[\left(\sum_{i,j=1}^{d}\left(W_{ij}^{(\kappa)}X_{ij}\right)^2\right)\cdot\left(\sum_{h,k=1}^{d}\left(W_{hk}^{(\kappa)}Y_{hk}\right)^2\right)\right]=$$
$$=\sum_{i,j,h,k=1}^{d}\mathbb{E}_{\mathcal{P}}\left[\left(W_{ij}^{(\kappa)}\right)^2\cdot(X_{ij})^2\cdot\left(W_{hk}^{(\kappa)}\right)^2\cdot(Y_{hk})^2\right]$$

and, finally,

$$\mathbb{E}_{\mathcal{P}}\left[\left(\sum_{i,j=1}^{d}\left(W_{ij}^{(\kappa)}X_{ij}\right)^2\right)\cdot\left(\sum_{h,k=1}^{d}\left(W_{hk}^{(\kappa)}Y_{hk}\right)^2\right)\right]=$$
$$=\sum_{i,j,h,k=1}^{d}\mathbb{E}_{\mathcal{P}}\left[\left(W_{ij}^{(\kappa)}\right)^2\cdot\left(W_{hk}^{(\kappa)}\right)^2\right](X_{ij})^2\cdot(Y_{hk})^2.$$

Due to the fact that $\mathcal{P}$ has zero mean[5], we obtain

$$\mathbb{E}_{\mathcal{P}}\left[\left(W_{ij}^{(\kappa)}\right)^2\cdot\left(W_{hk}^{(\kappa)}\right)^2\right]=m_4(\mathcal{P})\delta_{ih}\delta_{jk}$$

and this yields to

$$\mathbb{E}_{\mathcal{P}}\left[\left(\sum_{i,j=1}^{d}\left(W_{ij}^{(\kappa)}X_{ij}\right)^2\right)\cdot\left(\sum_{h,k=1}^{d}\left(W_{hk}^{(\kappa)}Y_{hk}\right)^2\right)\right]=m_4(\mathcal{P})\sum_{i,j=1}^{d}(X_{ij})^2(Y_{ij})^2. \tag{3.40}$$

Analogously, we can trait the remaining addends and obtain

$$\mathbb{E}_{n,\mathcal{P}}[\varphi_{kron-\pi}(\mathbf{X})\cdot\varphi_{kron-\pi}(\mathbf{Y})]=\sum_{n=0}^{\infty}\frac{\exp(-\frac{2}{\sigma^2})}{\sigma^{8n}}\frac{1}{\rho(n)(n!)^2}\cdot$$
$$\cdot\,9\prod_{\kappa=1}^{n}m_4(\mathcal{P})\sum_{i,j=1}^{d}(X_{ij})^2(Y_{ij})^2$$

and also

$$\mathbb{E}_{n,\mathcal{P}}[\varphi_{kron-\pi}(\mathbf{X})\cdot\varphi_{kron-\pi}(\mathbf{Y})]\leq\sum_{n=0}^{\infty}\frac{\exp(-\frac{2}{\sigma^2})}{\sigma^{8n}}\frac{9}{\rho(n)(n!)^2}\prod_{\kappa=1}^{n}m_4(\mathcal{P}) \tag{3.41}$$

due to the fact that

$$\sum_{i,j=1}^{d}(X_{ij})^2(Y_{ij})^2\leq\left(\sum_{i,j=1}^{d}(X_{ij})^2\right)\cdot\left(\sum_{i,j=1}^{d}(Y_{ij})^2\right)=\|\mathbf{X}\|_F^2\cdot\|\mathbf{Y}\|_F^2=1. \tag{3.42}$$

Thus, from equation (3.41), we get

$$\mathbb{E}_{n,\mathcal{P}}[\varphi_{kron-\pi}(\mathbf{X})\cdot\varphi_{kron-\pi}(\mathbf{Y})]\leq\sum_{n=0}^{\infty}\frac{1}{\rho(n)n!}\sum_{n=0}^{\infty}\frac{\exp(-\frac{2}{\sigma^2})}{\sigma^{8n}}\frac{1}{n!}\prod_{\kappa=1}^{n}9\,m_4(\mathcal{P}) \tag{3.43}$$

[5]In the case of the remaining 3 addends, in the very same way, we are able to compress all the index of the summations to just two by using the same property we are exploiting here.

because the product of two series upper bounds the series of the products. Hence, by definition of C_ρ,

$$\mathbb{E}_{n,\mathcal{P}}[\varphi_{\text{kron}-\pi}(\mathbf{X}) \cdot \varphi_{\text{kron}-\pi}(\mathbf{Y})] \leq C_\rho \exp(-\tfrac{2}{\sigma^2}) \sum_{n=0}^{\infty} \frac{9^n m_4(\mathcal{P})^n}{\sigma^{8n}} \frac{1}{n!}$$
$$= C_\rho \exp(-\tfrac{2}{\sigma^2}) \exp\left(\frac{9\, m_4(\mathcal{P})}{\sigma^8}\right)$$

and also

$$\mathbb{E}_{n,\mathcal{P}}[\varphi_{\text{kron}-\pi}(\mathbf{X}) \cdot \varphi_{\text{kron}-\pi}(\mathbf{Y})] \leq C_\rho \exp\left(\frac{9\, m_4(\mathcal{P}) - 2\sigma^4}{\sigma^8}\right) \tag{3.44}$$

We obtain (3.39) from (3.44) after the generality of $\mathbf{X}$ and $\mathbf{Y}$. □

If we neglect the function $\exp\left(\frac{9\, m_4(\mathcal{P})-2\sigma^4}{\sigma^8}\right)$, which is fixed after we select $\mathcal{P}$ and the bandwidth σ in (3.18), the boundary on the variance rewrites as C_ρ/ν^3. This means that, as the feature dimension ν increases, the variance very sharply converges to zero as $1/\nu^3$, i.e. our approximation converges to its expected value.

The constant C_ρ may however affect the quality of this limit. For instance, if we choose ρ to be a Geometric distribution of parameter $0 < \theta \leq 1$, we have $\rho(n) = (1-\theta)^n\theta$ and one can analytically obtain

$$C_\rho = \frac{1-\theta}{\theta} \exp\left(\frac{1-\theta}{\theta}\right). \tag{3.45}$$

The previous function increases and diverges for $\theta \to 1^-$ and $\theta \to 0^+$ making the bound potentially loose. The limit case $\theta \approx 0$ is very unfavorable also in in practice: in such a case a value sampled from ρ is high with high probability and, therefore, many Kronecker products need to be evaluated in (3.21). On the opposite side, the case $\theta \approx 1$ is very favorable in practical terms since n is small with high probability and therefore the cost of computing (3.21) approaches the minimal one. Further considerations on the practical choice of θ are also reported later on.

To conclude our discussion on the variance, we provide the following result, which is derived from Theorem 2 and 3 as a straightforward consequence of Chebyshev inequality.

Corollary 1. *Under the previous hypothesis, for any* $\epsilon > 0$ *and* $\mathbf{X}, \mathbf{Y}$ $d \times d$ *matrices, the probability* $\mathbb{P}\left[|\langle \boldsymbol{\Phi}_{\text{kron}-\pi}(\mathbf{X}), \boldsymbol{\Phi}_{\text{kron}-\pi}(\mathbf{Y})\rangle - K(\mathbf{X}, \mathbf{Y})| > \epsilon\right]$ *does not exceed the quantity* $\frac{C_\rho}{\nu^3\epsilon^2} \exp\left(\frac{9m_4(\mathcal{P})-2\sigma^4}{\sigma^8}\right)$.

This result ensures that the following probability

$$\mathbb{P}\left[|\langle \boldsymbol{\Phi}_{\text{kron}-\pi}(\mathbf{X}), \boldsymbol{\Phi}_{\text{kron}-\pi}(\mathbf{Y})\rangle - K(\mathbf{X}, \mathbf{Y})| > \epsilon\right]$$

is indeed small, since upper bounded by a quantity which is inversely quadratic in ϵ and inversely cubic in ν: this means that even a small value of ν ensures the latter probability to be small and guarantees the soundness of the approximation.

3.2.2 An alternative formulation

If inspecting equation (3.24), it would be natural to replace classical exponentiation - which works with scalars - with Kronecker exponentiation $\mathbf{X}^{\otimes n}$. However, with respect to the feature map $\boldsymbol{\phi}_{\text{kron}-\pi}$ presented in the previous Section, one may observe that $\otimes_{\kappa=1}^{n} \mathbf{W}^{(\kappa)\top}\mathbf{X} \neq \mathbf{X}^{\otimes n}$ for a general distribution of the weights (the equality would be true only if $\mathbf{W}^{(\kappa)}$ equals to the identity matrix $\mathbf{I}$ for every κ). We could thus argue that the following expression would be more appropriate for φ.

Definition 2. *Using the previous notations, for any* $d \times d$ *matrix* $\mathbf{X}$ *we define the scalar quantity*

$$\varphi_{\text{kron}-e}(\mathbf{X}) = \frac{1}{\sigma^{2n}} \sqrt{\frac{\exp(-\frac{1}{\sigma^2})}{\rho(n)n!}} \operatorname{tr}(\mathbf{V}^\top \mathbf{X}^{\otimes n}) \tag{3.46}$$

where $n \sim \rho$, *we still require* $\varphi_{\text{kron}-e}$ *to satisfy Assumption A.2 (see Theorem 2), while also assuming*

A′.1 The matrix $\mathbf{V}$ *is the Kronecker product of* n *matrices of size* $d \times d$, *whose entries are drawn independently from* $\mathcal{N}(0, \sigma^2)$ *(so are consequently the entries of* $\mathbf{V}$*).*

A.2 The $d \times d$ *matrix which is inputted to* $\varphi_{\text{kron}-e}$ *lies on the Frobenius norm-unitary sphere, that is* $\|\mathbf{X}\|_F = 1$.

Then, define the ν *dimensional vector* $\boldsymbol{\phi}_{\text{kron}-e}(\mathbf{X})$ *where each component is an independent realization of* $\varphi_{\text{kron}-e}(\mathbf{X})/\sqrt{\nu}$. *The explicit steps to compute* $\boldsymbol{\phi}_{\text{kron}-e}(\mathbf{X})$ *given* $\mathbf{X}$ *are enumerated in Algorithm 3.*

At a first glance, equation (3.46) seems closer to an arbitrary component of the exact feature map (3.24). This is because, as opposed to (3.21), the exponentiation operator for scalars is here directly replaced with the Kronecker exponentiation for matrices. Again, as for $\boldsymbol{\phi}_{\text{kron}-\pi}$, we introduce some random weights – here, denoted by $\mathbf{V}$ in order to accommodate for the compression generated by approximating an infinite dimensional vector.

For what concerns the assumptions, A.2 was also hypothesized in Section 12 and can be considered as a simple pre-processing step where each entry of the data $\mathbf{X}$ is divided by $\|\mathbf{X}\|_F$. On the contrary, if we compare A.1 with A'.1, we find a remarkable difference. In fact, A.1 was only constraining the mean and variance of the distribution $\mathcal{P}$. Differently, A'.1 not only constrains the probability distribution to be Gaussian but, additionally, we have to explicitly assume that $\mathbf{V}$ factorizes as the Kronecker product of n variables. Indeed, despite $\boldsymbol{\phi}_{\text{kron}-e}$ seems more naturally close to the exact feature map than $\boldsymbol{\phi}_{\text{kron}-\pi}$, it needs the more restrictive assumption A′.2. Without the latter, it is impossible to prove any theoretical result about the approximation $\langle \boldsymbol{\phi}_{\text{kron}-e}(\mathbf{X}), \boldsymbol{\phi}_{\text{kron}-e}(\mathbf{Y}) \rangle$ for (3.18).

It is straightforward to see that Definition 2 actually corresponds to the generalization to the kernel (3.18) of the approach in [Cav+17a], which is instead explicitly devised for the log-Euclidean kernel of covariance operators. Here, in fact, $\mathbf{X}$ and $\mathbf{Y}$ can be generic $d \times d$ data structures. Ultimately, we can state that

Algorithm 3: Approx, by Kronecker product

Input: A $d \times d$ input matrix $\mathbf{X}$, the desired feature size ν, the probability distributions ρ over integers and $\mathcal{P}$ over real numbers, the kernel bandwith $\sigma > 0$.

Output: $[\phi_{\text{kron-e},1}(\mathbf{X}), \dots, \phi_{\text{kron-e},\nu}(\mathbf{X})]$

foreach $j = 1, \dots, \nu$ **do**

  Sample n according to ρ

  Sample $\mathbf{V}$ as the Kronecker product of n random $d \times d$ matrices, each of the independently sampled from $\mathcal{P}$;

  Compute the scalar $e(\mathbf{X}) = \text{tr}\left(\mathbf{V}^\top \mathbf{X}^{\otimes n}\right)$

  Return $\phi_{\text{kron-e},j}(\mathbf{X}) = \sigma^{-2n}\left(\frac{\exp(-\sigma^{-2})}{\nu\rho(n)n!}\right)^{1/2} e(\mathbf{X})$

end

the approximation devised in [Cav+17a] is a particular case of $\boldsymbol{\Phi}_{\text{kron-e}}$, which, in turn, is a reformulation of $\boldsymbol{\Phi}_{\text{kron-}\pi}$. We can also prove what follows.

Theorem 4 (Unbiased approximation and bound on variance for $\boldsymbol{\Phi}_{\text{kron-e}}$). *Under the assumptions A′.1 and A.2, the linear kernel $\langle \boldsymbol{\Phi}_{\text{kron-e}}(\mathbf{X}), \boldsymbol{\Phi}_{\text{kron-e}}(\mathbf{Y})\rangle$ induced by $\varphi_{\text{kron-e}}$ is an unbiased estimator for $K(\mathbf{X}, \mathbf{Y}) = \exp(-\frac{1}{\sigma^2}\|\mathbf{X} - \mathbf{Y}\|_F^2)$. Actually it results*

$$\mathbb{E}_{n,\mathbf{V}}\left[\langle \boldsymbol{\Phi}_{\text{kron-e}}(\mathbf{X}), \boldsymbol{\Phi}_{\text{kron-e}}(\mathbf{Y})\rangle\right] = K(\mathbf{X}, \mathbf{Y}), \tag{3.47}$$

being the expected value jointly computed over all possible realizations of n from ρ and of the weight matrix $\mathbf{V}$.

In addition, the variance of the proposed estimator is explicitly bounded according to the following inversely-cubic function of ν, for C_ρ as in (3.35),

$$\text{var}_{n,\mathbf{V}}\left[\langle \boldsymbol{\Phi}_{\text{kron-e}}(\mathbf{X}), \boldsymbol{\Phi}_{\text{kron-e}}(\mathbf{Y})\rangle\right] \leq \frac{C_\rho}{\nu^3} \exp\left(\frac{3 - 2\sigma^2}{\sigma^4}\right). \tag{3.48}$$

As a corollary, for any $\mathbf{X}, \mathbf{Y}$ and $\epsilon > 0$,

$$\mathbb{P}\left[|\langle \boldsymbol{\Phi}_{\text{kron-e}}(\mathbf{X}), \boldsymbol{\Phi}_{\text{kron-e}}(\mathbf{Y}) - K(\mathbf{X}, \mathbf{Y})\rangle| > \epsilon\right] \leq \frac{C_\rho}{\nu^3\epsilon^2} \exp\left(\frac{3-2\sigma^2}{\sigma^4}\right). \tag{3.49}$$

Proof. Let $\mathbf{X}$ and $\mathbf{Y}$ be arbitrary $d \times d$ input matrices. Since $\boldsymbol{\Phi}_{\text{kron-e}}(\mathbf{X}) = [\phi_{\text{kron-e},1}(\mathbf{X}), \dots, \phi_{\text{kron-e},\nu}(\mathbf{X})]$ and the same happens for $\boldsymbol{\Phi}_{\text{kron-e}}(\mathbf{Y})$, combining the definition of Euclidean inner product and linearity of expectation gives

$$\mathbb{E}_{n,\mathcal{P}}[\langle \boldsymbol{\Phi}_{\text{kron-e}}(\mathbf{X}), \boldsymbol{\Phi}_{\text{kron-e}}(\mathbf{Y})\rangle] = \sum_{\ell=1}^{\nu} \mathbb{E}_{n,\mathcal{P}}[\phi_{\text{kron-e},j}(\mathbf{X}) \cdot \phi_{\text{kron-e},j}(\mathbf{Y})] \tag{3.50}$$

and, we also obtain

$$\mathbb{E}_{n,\mathcal{P}}[\langle \boldsymbol{\Phi}_{\text{kron-e}}(\mathbf{X}), \boldsymbol{\Phi}_{\text{kron-e}}(\mathbf{Y})\rangle] = \sum_{\ell=1}^{\nu} \mathbb{E}_{n,\mathcal{P}}[\varphi_{\text{kron-e}}(\mathbf{X})/\sqrt{\nu} \cdot \varphi_{\text{kron-e}}(\mathbf{Y})/\sqrt{\nu}] \tag{3.51}$$

applying the definition of the map $\varphi_{\text{kron-e}}$. Clearly,

$$\mathbb{E}_{n,\mathcal{P}}[\langle \boldsymbol{\phi}_{\text{kron-e}}(\mathbf{X}), \boldsymbol{\phi}_{\text{kron-e}}(\mathbf{Y})\rangle] = \sum_{\ell=1}^{\nu} \tfrac{1}{\nu}\mathbb{E}_{n,\mathcal{P}}[\varphi_{\text{kron-e}}(\mathbf{X}) \cdot \varphi_{\text{kron-e}}(\mathbf{Y})] = \\ = \mathbb{E}_{n,\mathcal{P}}[\varphi_{\text{kron-e}}(\mathbf{X}) \cdot \varphi_{\text{kron-e}}(\mathbf{Y})]$$

by using the linearity of expectation and the fact that, in the previous summation over ℓ, all addends are equal.

Write down the explicit formulas for $\varphi_{\text{kron-e}}(\mathbf{X})$ and $\varphi_{\text{kron-e}}(\mathbf{Y})$ and expand the expectation over ρ.

$$\mathbb{E}_{n,\mathcal{P}}[\langle \boldsymbol{\phi}_{\text{kron-e}}(\mathbf{X}), \boldsymbol{\phi}_{\text{kron-e}}(\mathbf{Y})\rangle] \\ = \exp(-\tfrac{1}{\sigma^2}) \sum_{n=0}^{\infty} \frac{1}{\sigma^{2n}} \frac{1}{n!} \mathbb{E}_{\mathcal{P}} \left[\text{tr}\left(\mathbf{V}^\top \mathbf{X}^{\otimes n}\right) \text{tr}\left(\mathbf{V}^\top \mathbf{Y}^{\otimes n}\right)\right]. \tag{3.52}$$

Now, let us inspect the term $\text{tr}\left(\mathbf{V}^\top \mathbf{X}^{\otimes n}\right)$ separately. For better understanding the next step, let us notice that a generic entry of the matrix $\mathbf{X}^{\otimes n} = \mathbf{X} \otimes \cdots \otimes \mathbf{X}$ is composed by products of the type $X_{r_1,c_1} \cdot X_{r_2,c_2} \cdots X_{r_n,c_n}$, where, for each $\ell = 1, \ldots, n$, the row- and column-indexes $r_\ell = r_\ell(i,j)$ and $c_\ell = c_\ell(i,j)$ do depend upon the position (i,j) of the entry of $\mathbf{X}^{\otimes n}$ that we are interested in computing. Therefore, due to the property $\text{tr}(\mathbf{A}^\top \mathbf{B}) = \langle \mathbf{A}, \mathbf{B}\rangle_F := \sum_{ij} A_{ij} \cdot B_{ij}$, we get

$$\langle \mathbf{V}, \mathbf{X} \otimes \cdots \otimes \mathbf{X}\rangle_F = \sum_{i,j=1}^{d} V_{ij} [\mathbf{X} \otimes \cdots \otimes \mathbf{X}]_{ij} = \prod_{\ell=1}^{n} \sum_{i,j=1}^{d} V_{r_\ell(i,j),c_\ell(i,j)} X_{r_\ell(i,j),c_\ell(i,j)} \tag{3.53}$$

where, thanks to the assumption A'.1, $\mathbf{V}$ is factorizes over the Kronecker product of n matrices – that we still call $\mathbf{V}$ for the sake of simplicity – each of them identically sampled entrywise from $\mathcal{P}$.

By merging (3.53) in (3.52), thanks to the properties of the expected value, we rewrite $\mathbb{E}_{n,\mathcal{P}}[\langle \boldsymbol{\phi}_{\text{kron-e}}(\mathbf{X}), \boldsymbol{\phi}_{\text{kron-e}}(\mathbf{Y})\rangle]$ as

$$\exp(-\tfrac{1}{\sigma^2}) \sum_{n=0}^{\infty} \frac{1}{\sigma^{2n}} \frac{1}{n!} \prod_{\ell=1}^{n} \sum_{i,j,h,k=1}^{d} \mathbb{E}_{\mathcal{P}} \left[V_{r_\ell(i,j),c_\ell(i,j)} V_{r_\ell(h,k),c_\ell(h,k)} \cdot \right. \\ \left. \cdot X_{r_\ell(i,j),c_\ell(i,j)} Y_{r_\ell(h,k),c_\ell(h,k)} \right]$$

or, equivalently,

$$\exp(-\tfrac{1}{\sigma^2}) \sum_{n=0}^{\infty} \frac{1}{\sigma^{2n}} \frac{1}{n!} \prod_{\ell=1}^{n} \sum_{i,j,h,k=1}^{d} \mathbb{E}_{\mathcal{P}} \left[V_{r_\ell(i,j),c_\ell(i,j)} V_{r_\ell(h,k),c_\ell(h,k)} \right] \cdot \tag{3.54}$$

$$X_{r_\ell(i,j),c_\ell(i,j)} Y_{r_\ell(h,k),c_\ell(h,k)} \tag{3.55}$$

due to the properties of the expected value with respect to the product of independent random variables.

Due to the assumption that the entries in $\mathbf{V}$ are all identically and independently distributed according to the distribution $\mathcal{P}$, we get

$$\mathbb{E}_{\mathcal{P}}\left[V_{rc}V_{r'c'}\right] = \mathrm{var}[\mathcal{P}]\delta_{rc}\delta_{r'c'} \tag{3.56}$$

by exploiting assumption A.1 that ensures $\mathbb{E}[\mathcal{P}] = 0$. If we implement (3.56) into (3.55), we obtain

$$\begin{aligned}
\mathbb{E}_{n,\mathcal{P}}[\langle \boldsymbol{\phi}_{\mathsf{kron-e}}(\mathbf{X}), \boldsymbol{\phi}_{\mathsf{kron-e}}(\mathbf{Y})\rangle] &= \exp(-\tfrac{1}{\sigma^2}) \sum_{n=0}^{\infty} \frac{1}{\sigma^{2n}} \frac{1}{n!} \prod_{\ell=1}^{n} \sum_{i,j=1}^{d} X_{ij}Y_{ij} = \\
&= \exp(-\tfrac{1}{\sigma^2}) \sum_{n=0}^{\infty} \frac{1}{\sigma^{2n}} \frac{1}{n!} \prod_{\ell=1}^{n} \langle \mathbf{X}, \mathbf{Y}\rangle = \\
&= \exp(-\tfrac{1}{\sigma^2}) \sum_{n=0}^{\infty} \left(\frac{\langle \mathbf{X}, \mathbf{Y}\rangle}{\sigma^2}\right)^n \frac{1}{n!} = \\
&= \exp(-\tfrac{1}{\sigma^2}) \exp\left(\frac{\langle \mathbf{X}, \mathbf{Y}\rangle}{\sigma^2}\right) = \\
&= \exp\left(\frac{\langle \mathbf{X}, \mathbf{Y}\rangle - 1}{\sigma^2}\right) = \\
&= \exp\left(-\frac{2 - 2\langle \mathbf{X}, \mathbf{Y}\rangle}{2\sigma^2}\right)
\end{aligned} \tag{3.57}$$

where, in the third-to-last stage, we applied the Taylor expansion for the exponential function. Now, using assumption A.3, we can substitute 2 in (3.57) with $1+1 = \|\mathbf{X}\|_F^2 + \|\mathbf{Y}\|_F^2$, yielding to

$$\begin{aligned}
\mathbb{E}_{n,\mathcal{P}}[\langle \boldsymbol{\phi}_{\mathsf{kron-e}}(\mathbf{X}), \boldsymbol{\phi}_{\mathsf{kron-e}}(\mathbf{Y})\rangle] &= \exp\left(-\frac{\|\mathbf{X}\|_F^2 - 2\langle \mathbf{X}, \mathbf{Y}\rangle + \|\mathbf{Y}\|_F^2}{2\sigma^2}\right) = \\
&= \exp\left(-\frac{\|\mathbf{X} - \mathbf{Y}\|_F^2}{2\sigma^2}\right) = \\
&= K(\mathbf{X}, \mathbf{Y}).
\end{aligned} \tag{3.58}$$

Similarly, to prove the bound on the variance, by definition of inner product and linearity of the variance with respect to sum of independent random variables, we get

$$\mathrm{var}_{n,V}\left[\langle \boldsymbol{\phi}_{\mathsf{kron-e}}(\mathbf{X}), \boldsymbol{\phi}_{\mathsf{kron-e}}(\mathbf{Y})\rangle\right] = \sum_{\ell=1}^{\nu} \mathrm{var}_{n,V}\left[\phi_{\mathsf{kron-e},\ell}(\mathbf{X}) \cdot \phi_{\mathsf{kron-e},\ell}(\mathbf{Y})\right] \tag{3.59}$$

and

$$\mathrm{var}_{n,V}\left[\langle \boldsymbol{\phi}_{\mathsf{kron-e}}(\mathbf{X}), \boldsymbol{\phi}_{\mathsf{kron-e}}(\mathbf{Y})\rangle\right] = \sum_{\ell=1}^{\nu} \mathrm{var}_{n,V}\left[\varphi_{\mathsf{kron-e}}(\mathbf{X})/\sqrt{\nu} \cdot \varphi_{\mathsf{kron-e}}(\mathbf{Y})/\sqrt{\nu}\right] \tag{3.60}$$

due to the construction of each component of $\boldsymbol{\phi}_{\mathsf{kron-e}}$ as independent realization of $\varphi_{\mathsf{kron-e}}$. Thanks to the fact that, for every random variable Z, it results

$\text{var}[Z] \leq \mathbb{E}[Z^2]$, taking advantage of the property of expectation, we get

$$\text{var}_{n,\mathbf{V}}\left[\langle \boldsymbol{\Phi}_{\text{kron}-e}(\mathbf{X}), \boldsymbol{\Phi}_{\text{kron}-e}(\mathbf{Y})\rangle\right] \leq \frac{1}{\nu^3}\mathbb{E}_{n,\mathbf{V}}\left[\varphi_{\text{kron}-e}(\mathbf{X}) \cdot \varphi_{\text{kron}-e}(\mathbf{Y})\right] \tag{3.61}$$

and, by explicitly using the definition of $\varphi_{\text{kron}-e}$, we achieve

$$\begin{aligned}&\text{var}_{n,\mathbf{V}}\left[\langle \boldsymbol{\Phi}_{\text{kron}-e}(\mathbf{X}), \boldsymbol{\Phi}_{\text{kron}-e}(\mathbf{Y})\rangle\right] \\ &\quad\leq \frac{1}{\nu^3}\sum_{n=0}^{\infty}\rho(n)\frac{1}{\sigma^{8n}}\left(\frac{\exp(-\frac{1}{\sigma^2})}{\rho(n)n!}\right)^2 \mathbb{E}_{\mathbf{V}}\left[\text{tr}(\mathbf{V}^\top\mathbf{X}^{\otimes n})^2\text{tr}(\mathbf{V}^\top\mathbf{Y}^{\otimes n})^2\right]\end{aligned}$$

and

$$\begin{aligned}&\text{var}_{n,\mathbf{V}}\left[\langle \boldsymbol{\Phi}_{\text{kron}-e}(\mathbf{X}), \boldsymbol{\Phi}_{\text{kron}-e}(\mathbf{Y})\rangle\right] \\ &\quad\leq \frac{C_\rho}{\nu^3}\exp(-\tfrac{2}{\sigma^2})\sum_{n=0}^{\infty}\frac{1}{\sigma^{8n}}\frac{1}{n!}\mathbb{E}_{\mathbf{V}}\left[\text{tr}(\mathbf{V}^\top\mathbf{X}^{\otimes n})^2\text{tr}(\mathbf{V}^\top\mathbf{Y}^{\otimes n})^2\right]\end{aligned}$$

since we exploited the property that the product of two series gives an upper bound for the series of the term-by-term products. By means of (3.53) and the notation thereby introduced, we can upper bound $\text{var}_{n,\mathbf{V}}\left[\langle \boldsymbol{\Phi}_{\text{kron}-e}(\mathbf{X}), \boldsymbol{\Phi}_{\text{kron}-e}(\mathbf{Y})\rangle\right]$ by

$$\begin{aligned}&\frac{C_\rho}{\nu^3}\exp(-\tfrac{2}{\sigma^2})\sum_{n=0}^{\infty}\frac{1}{\sigma^{8n}}\frac{1}{n!}\prod_{\ell=1}^{n}\mathbb{E}_{\mathbf{V}}\left[\left(\sum_{i,j=1}^{d}V_{r_\ell(i,j),c_\ell(i,j)}X_{r_\ell(i,j),c_\ell(i,j)}\right)^2\right]\cdot \\ &\qquad\cdot\mathbb{E}_{\mathbf{V}}\left[\left(\sum_{h,k=1}^{d}V_{r_\ell(h,k),c_\ell(h,k)}Y_{r_\ell(h,k),c_\ell(h,k)}\right)^2\right],\end{aligned} \tag{3.62}$$

where, for notational simplicity, we still use the same letter to denote the factors in which $\mathbf{V}$ is assumed to be factorized into.

But now, repeating the same steps which yields to obtain (3.40),

$$\text{var}_{n,\mathbf{V}}\left[\langle \boldsymbol{\Phi}_{\text{kron}-e}(\mathbf{X}), \boldsymbol{\Phi}_{\text{kron}-e}(\mathbf{Y})\rangle\right] \leq \frac{C_\rho}{\nu^3}\exp(-\tfrac{2}{\sigma^2})\sum_{n=0}^{\infty}\frac{1}{\sigma^{8n}}\frac{1}{n!}3^n\sigma^{4n}\left(\langle\mathbf{X},\mathbf{Y}\rangle_F\right)^n \tag{3.63}$$

in which we can substitute the symbol forth order momentum with its explicit value that is directly computable for a Gaussian distribution. By using the Taylor expansion of the exponential function, we obtain

$$\begin{aligned}\text{var}_{n,\mathbf{V}}\left[\langle \boldsymbol{\Phi}_{\text{kron}-e}(\mathbf{X}), \boldsymbol{\Phi}_{\text{kron}-e}(\mathbf{Y})\rangle\right] &\leq \frac{C_\rho}{\nu^3}\exp(-\tfrac{2}{\sigma^2})\exp\left(\frac{3\langle\mathbf{X},\mathbf{Y}\rangle_F}{\sigma^4}\right) \\ &\leq \frac{C_\rho}{\nu^3}\exp(-\tfrac{2}{\sigma^2})\exp\left(\frac{3}{\sigma^4}\right)\end{aligned}$$

because of (3.42). The thesis follows from rearranging terms in the previous relationship and due to the generality of $\mathbf{X}, \mathbf{Y}$.

Finally, thanks to Chebyshev inequality, the last results is a direct consequence of the unbiasedness and the bound of the variance. □

Computational cost

Interestingly, we can observe one common trend which is shared across all the approaches [RR07; Vem+10; VZ12; KK12; Cav+17a]: in computational terms, the number of products required for computing one component of the feature map is linear with respect to the data dimensionality (which is $O(d^2)$ since log-covariance $d \times d$ matrices are used as input). Among the previously published works, two papers are different: [Le+13] achieves a log-linear complexity, while, unfortunately, [Rin] has exponential complexity with respect to the data size: this is the reason why we were not able to include [Rin] among the methods in comparison.

We can thus observe that the cost of calculating

$$\mathrm{tr}(\otimes_{\kappa=1}^{n} \mathbf{W}^{(\kappa)\top}\mathbf{X}) = \prod_{\kappa=1}^{n} \mathrm{tr}(\mathbf{W}^{(\kappa)\top}\mathbf{X})$$

(in the computation of $\boldsymbol{\phi}_{\mathsf{kron}-\pi}$) is linear in both the input data dimensionality and in n. Similarly, the same holds for $\boldsymbol{\phi}_{\mathsf{kron}-\mathsf{e}}$, thanks to the factorization assumption A′.1.

Despite such linear dependence from n may appear as a drawback, we can take advantage of the freedom in choosing ρ in order to keep n small. Indeed, throughout all the experiments, either involving $\boldsymbol{\phi}_{\mathsf{kron}-\pi}$ or $\boldsymbol{\phi}_{\mathsf{kron}-\mathsf{e}}$, we fixed ρ as a Geometrical distribution of parameter $\theta = 0.9$. This ensures that the probability of sampling high values of n from ρ is practically zero. Indeed, through analytical computations, we can also notice that, for each realization of $\varphi_{\mathsf{kron}-\pi}$ or $\varphi_{\mathsf{kron}-\mathsf{e}}$, $\mathbb{P}(n > 3) = 0.04$.

This makes the computational cost of our approach substantially in line with that of other works[RR07; Vem+10; VZ12; KK12; Cav+17a].

Analysis of action recognition performance

We present here all the datasets considered for the experiments: UTKinect [Xia+12], Florence3D [Sei+13], MSR-Action-Pairs (MSR-pairs) [OL13], MSR-Action3D [Li+10b], Gaming-3D (G3D) [Blo+12], HDM-05 [M+07], MSRC- Kinect12 [Fot+12] and NTU RGB+D [Sha+16]. Table 3.3 summarizes statistics for each.

We follow usual training and testing splits proposed in the literature. For Florence3D, G3D, and UTKinect, we use the protocols of [Vem+14; VC16; Zha+16a]. For MSR-Action3D, we adopt the splits originally proposed by [Li+10b]. On MSRC-Kinect12, once highly corrupted action instances are removed as in [Hus+13], training is performed on odd-index subject, while testing on the even-index ones. On HDM-05, the training split exploits all the data from the "bd" and "mm" subjects, being "bk", "dg" and "tr" left out for testing [Wan+15b]. To be consistent with the literature, we replicated the 14 classes experiments (HDM-05_{14}) as in [Wan+15b; Cav+16]. When dealing with the whole dataset (HDM-05_{all}), since some of the total classes are missing from the training/testing splits, we adopted the protocol of [CC14] to partition the dataset into 65 action classes. For NTU RGB+D, we followed the authors' instruction [Sha+16] in removing the most corrupted instances, also purging

	Classes	Subjects	Repetitions	Samples	Joints
UTKinect [Xia+12]	10	10	1-2	199	20
Florence3D [Sei+13]	9	10	2-3	215	15
MSR-pairs [OL13]	20	10	1-3	353	20
MSR-Action3D [Li+10b]	20	10	2-3	567	20
G3D [Blo+12]	20	10	2-4	663	20
HDM-05_{14} [M̈+07]	14	5	8-10	686	31
HDM-05_{all} [M̈+07]	65	5	8-40	2343	31
MSRC-Kinect12 [Fot+12]	12	30	10-25	5881	20
NTU-$\times$*subject* [Sha+16] NTU-$\times$*view* [Sha+16]	60	40	80	56578	25

TABLE 3.3: Statistics about the considered datasets.

the trials with missing joints recordings. Finally, we replicated both the cross-subject and cross-view testing protocols proposed in [Sha+16], denoting them as NTU-$\times$-subject and NTU-$\times$-*view*.

In all experiments, as a common data pre-processing step [Vem+14; PCSC14; VC16; Zha+16a; Sha+16; Cav+16; Kon+16; Liu+16], we fix one root joint (the one located at the hip center), and we compute the relative differences of all the other $J-1$ 3D joint positions. By doing this at any timestamps $t = 1, \dots, T$ we obtain a $3(J-1)$-dimensional (column) vector $\mathbf{p}(t)$ of relative displacements. As the representation for data instance $[\mathbf{p}(1), \dots, \mathbf{p}(T)]$, we compute a covariance matrix

$$\mathbf{C} = \frac{1}{T-1} \sum_{t=1}^{T} (\mathbf{p}(t) - \boldsymbol{\mu})(\mathbf{p}(t) - \boldsymbol{\mu})^\top, \tag{3.64}$$

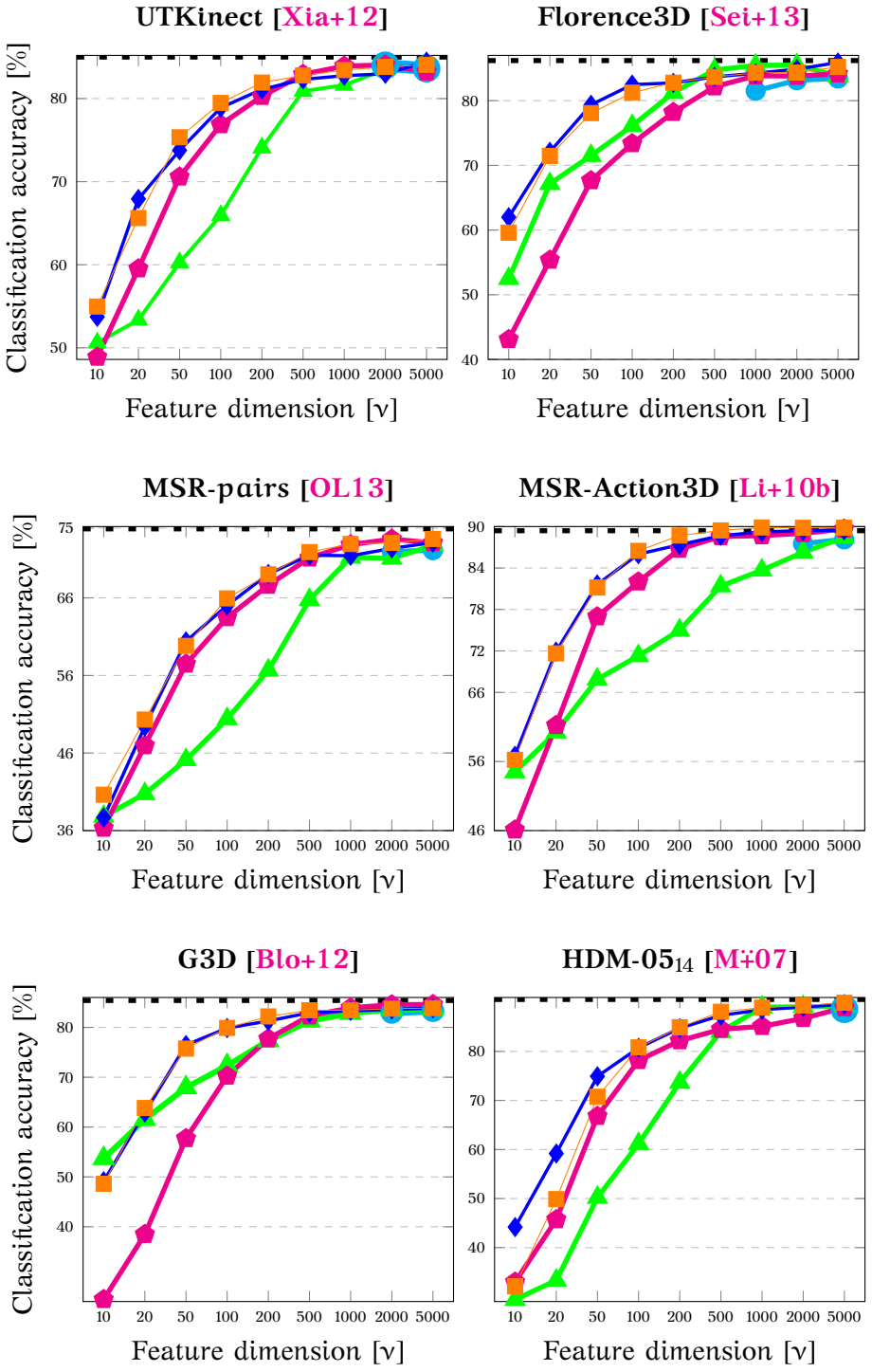
being $\boldsymbol{\mu} = \frac{1}{T} \sum_{t=1}^{T} \mathbf{p}(t)$ the temporal average of $\mathbf{p}(t)$. Finally, the input representation for our approximated feature map is obtained as

$$\mathbf{X} = \log \mathbf{C} = \mathbf{U} \operatorname{diag}(\log(\boldsymbol{\sigma}))\mathbf{U}^\top, \tag{3.65}$$

being $\boldsymbol{\sigma}$ the vector of eigenvalues (eventually regularized by an additive factor as in [Min+14b]) and $\mathbf{U}$ the matrix of eigenvectors of $\mathbf{C}$. Finally, since the log of a symmetric matrix is symmetric, in order to avoid to process identical entries twice, we zero out all the lower triangular entries in $\mathbf{X}$ and we divide all of them by $\|\mathbf{X}\|_F$.

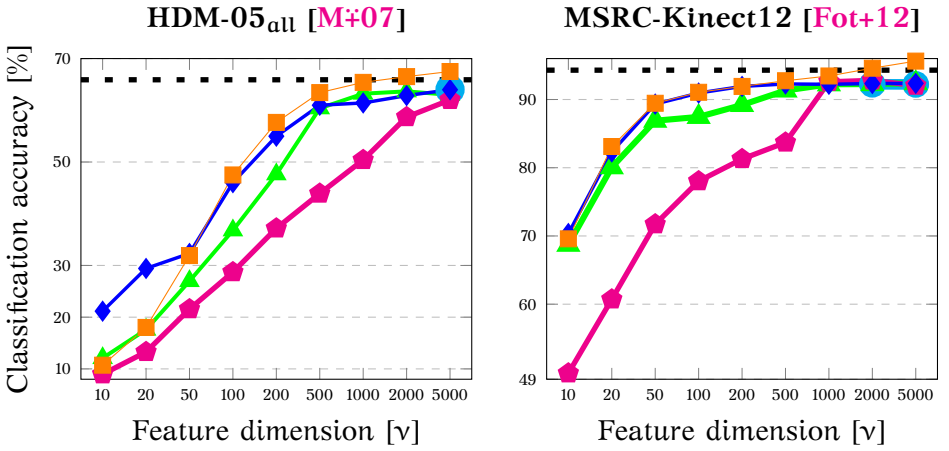
Despite the several approximations [RR07; Vem+10; VZ12; Le+13; KK12; Rin; Cav+17a] have been proposed and are applicable to a RBF kernel function (3.18), to the best of our knowledge there is no clear evidence of which method is effective and efficient for classification. Indeed, despite all methods ensure scalability in the big data regime, there is no clear understanding about which method gives superior performance and, in general, how a good feature dimensionality ν should be chosen in practice. Here, we try to answer this question with a detailed analysis of 3D action recognition accuracies on the 10 benchmark datasets listed in Table 3.3, while the feature dimensionality ν assumes one of the following values: 10, 20, 50, 100, 200, 500, 1000, 2000, 5000. We report the results of this analysis in Figures 3.5, 3.6 and 3.7 and we will discuss the scored results in the following.

If we compare all the methods analyzed in Figures 3.5, 3.6 and 3.7, we can find a common behavior, that is a growth of accuracy while ν increases. This it theoretically reasonable because $\boldsymbol{\phi}_{\text{kron}-\pi}$ and $\boldsymbol{\phi}_{\text{kron}-e}$, as well as the alternative



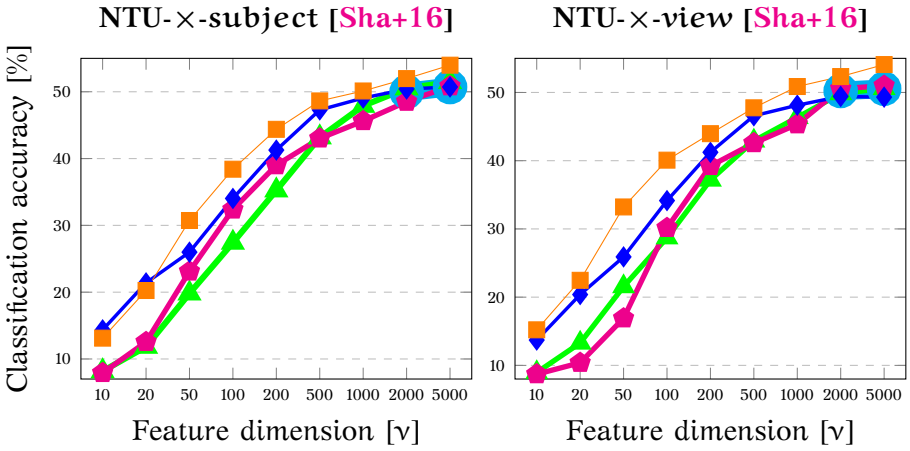
Hadamard approx [Le+13] (cyan), Fourier approx [RR07; Vem+10; VZ12] (green), Taylor approx [KK12] (magenta),
$\Phi_{\text{kron}-\text{e}}$ *(proposed, blue)*, $\Phi_{\text{kron}-\pi}$ *(proposed, orange)*.

FIGURE 3.5: Small data regime ($\sim 10^2$ samples). Experiments on the UTKinect, Florence 3D, MSR-pairs, MSR-Action3D, G3D and HDM-05 using the selection of 14 classes used by [Hus+13]. In each case, we monitor the changes in action recognition accuracy as a function of the feature dimensionality ν. Across figures, the same color refers to same method. Best viewed in color.



Hadamard approx [Le+13] (cyan), Fourier approx [RR07; Vem+10; VZ12] (green), Taylor approx [KK12] (magenta),
$\Phi_{\text{kron}-e}$ *(proposed, blue)*, $\Phi_{\text{kron}-\pi}$ *(proposed, orange)*.

FIGURE 3.6: Medium data regime ($\sim 10^3$ samples). Experiments on the full HDM-05 datasets and on the MSRC-Kinect 12. In each case, we monitor the changes in action recognition accuracy as a function of the feature dimensionality ν. Across figures, the same color refers to same method. Best viewed in color.



Hadamard approx [Le+13] (cyan), Fourier approx [RR07; Vem+10; VZ12] (green), Taylor approx [KK12] (magenta),
$\Phi_{\text{kron}-e}$ *(proposed, blue)*, $\Phi_{\text{kron}-\pi}$ *(proposed, orange)*.

FIGURE 3.7: Big data regime ($\sim 10^4$ samples). Experiments on the NTU RGB+D dataset [Sha+16] adopting either the cross-subject or the cross-view protocol. In each case, we monitor the changes in action recognition accuracy as a function of the feature dimensionality ν. Across figures, the same color refers to same method. *Note that, differently from Figures 3.5 and 3.6, the size of the dataset does not allow to directly train a kernel machine, and approximated schemes are obliged.* Best viewed in color.

methods [RR07; Vem+10; VZ12; Le+13; KK12; Rin; Cav+17a] are guaranteed to provide a better approximation for a bigger ν. In our case, the reason is that the bound on the variance is $O(1/\nu^3)$.

As another piece of evidence for correctness of the approximations, we can notice that with a feature dimensionality $\nu \geq 1000$, the performance of each single method is close to the remaining ones and, globally, they are able to mimic the classification accuracy of an exact kernel machine trained in either the small (UTKinect, Florence3D, MSR-pairs, MSR-Action3D, G3D) or medium (HDM-05 and MSRC-Kinect12) data regime. The previous claim is also corroborated from the fact that, in those datasets, when $\nu > 500, 1000$, we observe a plateau of accuracies since all methods tend to approach the horizontal asymptote given by the exact kernel method (black dotted line). Differently, while moving to the bigger NTU RGB+D, the previous plateau disappears, meaning that an additional increase in the feature dimensionality could be beneficial for improving action recognition.

However, we can observe an interesting pattern which is, in general, common to all datasets for the case $\nu < 200$: at low feature dimensionality (such as 10 or 20), the proposed approximations $\boldsymbol{\phi}_{\text{kron}-\pi}$ and $\boldsymbol{\phi}_{\text{kron}-\text{e}}$ are remarkably superior in performance with respect to all other competitors which are outperformed by margin. For instance, +10% on Florence3D for $\nu = 10$, +14% on MSR-Action3D for $\nu = 20$, +9% on G3D when $\nu = 50$ and more than +10% on HDM-05_{all} when $\nu = 100$. Moreover, if comparing $\boldsymbol{\phi}_{\text{kron}-\pi}$ and $\boldsymbol{\phi}_{\text{kron}-\text{e}}$, we can observe that, in the small data regime (Figure 3.5) the two methods are more or less equivalent. Instead, in the middle and big data regime (Figures 3.6 and 3.7), $\boldsymbol{\phi}_{\text{kron}-\text{e}}$ is systematically outperformed by $\boldsymbol{\phi}_{\text{kron}-\pi}$.

Finally, as anticipated, an interesting collateral result of our work consists in the possibility to compare the previously proposed methods [RR07; Vem+10; VZ12; Le+13; KK12] within a common benchmark. Indeed, despite [Le+13] shows a solid performance which is always able to match the exact kernel machine and all the other competitors, such approach is limited by the impossibility to obtain a low-dimensional feature representation. Differently, the Fourier [RR07; Vem+10; VZ12] and Taylor-based methods [KK12] show an oscillating performance where, frequently, one outperforms the other, even by margin. In this respect, the solidity of our methods, which is always top scoring, can be concretely appreciated as an advantage.

3.3 Learning how to weight joints' correlations: Log-COV-Net

Covariance-based representations for action recognition from skeletal joints have attested a superior performance [Hus+13; Har+14; Wan+15b; Min+16b; Cav+16]. However, in order to fully exploit the structure which is induced by the covariance representation, classifiers have to be kernelized in order to fully exploit Riemannian geometry when learning decision boundaries to discriminate across different actions. Despite this being mathematically fine, some computational drawbacks arise. Indeed, training such a classifier often requires the computation of Gram matrices, whose quadratic complexity in

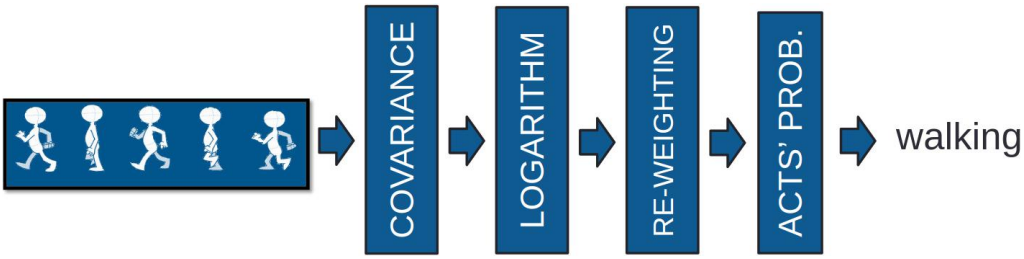


FIGURE 3.8: Log-COV-net architecture. When the kinematics is properly modeled, there is no need for the architecture to be deep in order to achieve a effective action recognition performance. We verify this claim by proposing a very shallow neural network explained in the figure: we compute a temporal covariance representation that is log-projected and afterwards re-weighted before entering into the final classification layer. Essentially, we use a data-driven approach to rescale the importance of the temporal correlation so that only the most important ones ultimately lead the recognition stage.

terms of data instances makes the whole procedure intractable in the big data regime.

On the other side, feature learning approaches via neural networks fully benefit from a gigantic amount of training examples to optimize the huge number (millions, billions) of parameters present in a deep network. At the same time, this is the main reason for the astonishing results scored by data-representations and the source of difficulty in effectively training such networks. Indeed, the optimization problem is non-convex, prone to overfitting, requiring acceleration through parallel GPU computation.

Therefore, despite the strong performance provided by either covariance-based or feature learning paradigms, each of them has its own drawbacks (scalability versus difficult training, respectively). To this end, in this work we aim at intertwining covariance-based and feature approaches in order to combine their pros and get rid of the cons. Namely, our unifying approach will achieve state-of-the-art classification, guaranteeing scalability to the big data regime and allowing easy and fast training/inference on CPU. This is possible by leveraging our intuition that, since exploiting the powerful covariance representation to encode action dynamics, there is no need for the network to be deep. In fact, shallow architectures are just enough in mining discriminative patterns for action classification.

We now present the proposed approach called Log-Covariance Network, which is sketched in Fig. 3.8, and we provide and intuition for it. For each action instance a, acquired in the form of the multi-dimensional time series (2.6), we compute a covariance matrix a according to formula (2.3). Then, we project $\mathbf{X}_a$ by a logarithm mapping log. By exploiting the eigendecomposition

$$\mathbf{X}_a = \mathbf{U} \begin{bmatrix} \lambda_1 & 0 & \dots & 0 \\ 0 & \lambda_2 & \dots & 0 \\ & & \ddots & \\ 0 & 0 & \dots & \lambda_{3J} \end{bmatrix} \mathbf{U}^\top \tag{3.66}$$

for $\mathbf{X}_a$, $\log \mathbf{X}_a$ is trivial to compute in as follows

$$\log \mathbf{X}_a = \mathbf{U} \begin{bmatrix} \log \lambda_1 & 0 & \dots & 0 \\ 0 & \log \lambda_2 & \dots & 0 \\ & & \ddots & \\ 0 & 0 & \dots & \log \lambda_{3J} \end{bmatrix} \mathbf{U}^\top, \tag{3.67}$$

since all λ_i are strictly positive. Formally, this is interpreted as a projection over the tangent space [Har+14], which is locally Euclidean, naturally inducing a vectorization which does not corrupt the geometry. Precisely, we define $\mathbf{v}_a$ to be the vectorization of all diagonal and lower-diagonal entries[6] of $\log \mathbf{X}_a$: as similarly done in [Hus+13; Wan+15b; Cav+16] such intermediate representation is fully able to provide and Euclidean (vectorial) representation which keeps the powerfulness of the Riemannian encoding as SPD matrix [Har+14]. Finally, the vector $\mathbf{v}_a$ is fed into a fully connected (FC) layer, followed by a sigmoid linearity, which is in turn fed into a classification layer where a hinge loss is exploited. We call our network *Log-COV-Net*. Despite all matrices $\mathbf{X}_a$ are positive definite in theory, due to numerical issues, the computed eigenvalues are not always positive: before applying the log mapping, we replace λ_i with $\lambda_i' = \lambda_i + 10^{-4}$. With respect to Fig. 3.8, note that the "covariance" and "logarithm" layers (which implement equation (2.3) and (3.67), respectively) are parameter-free. The only parameter to be trained are the weights $\mathbf{W}$ of the fully connected layer and, of course, the ones of the final classification layer. In our experimental setup, we found that if we jointly train $\mathbf{W}$ and the classifier's parameters, we are highly sensitive to the size of the FC layer. Differently, we achieve more stability by pre-training the FC weights with a cross-entropy loss, also exploiting the powerfulness of supervision. For doing that, we use conjugate gradient descent for all experiments except the ones on NTU-RGB+D [Sha+16] dataset where we exploit ADAM optimizer with mini-batches of 1024 elements. As a final step, we separately train the hinge-loss classification layer.

We can provide a solid theoretical background for the architecture we just described. In order to do so, we will consider the approximated feature map $\boldsymbol{\phi}_{\text{kron}-\pi}$ and $\boldsymbol{\phi}_{\text{kron}-\text{e}}$ that we introduced in Sections 3.2.1 and 3.2.2. In fact, for both, in addition to some sort of random weighting, another source of randomness is present: the integer n. As explained in Section 3.2.1, n builds upon the formal analogy with the exact feature map associated to the exact RBF kernel function obtained through Taylor expansion. In fact, n can be related to the degree of the aforementioned expansion in the sense, as opposed to considering all the - exponentially growing - terms of an arbitrary degree n, we select only *one* of those terms, the latter being randomly rescaled in order to recover from this compression.

In this Section we will consider the case where, instead of randomly selecting n (from a Geometrical distribution), we deterministically fix it to be $n = 1$. First of all, let us observe that this makes (3.21) and (3.46) formally identical and corresponds selecting only the component of degree 1 in (3.24). In these terms, we can interpret it as a *linearization* of the exact feature map associated to the RBF kernel function.

[6] Due to simmetry, the upper-diagonal elements are the same as the lower-diagonal ones

Intuitively, the randomness in n can lead to "explore" all the infinite components in the exact feature map $\mathbf{f}(\mathbf{X})$ in (3.24) in order to accumulate enough patterns in $\boldsymbol{\phi}_{\text{kron}-\pi}$ and $\boldsymbol{\phi}_{\text{kron}-e}$ to properly approximate the RBF Gaussian kernel. Imposing $n = 1$ can be instead thought of as a sort of linearization as to approximate $\mathbf{f}(\mathbf{X})$ in (3.24). In such a case, there is clearly a little room for the random weights to help recovering from the compression. Therefore, as an opposed paradigm to randomly sample the weights, we can try to learn them in a data-driven fashion, in order to promote class-disambiguation. In fact, since our ultimate goal is accomplishing the action recognition task, the perspective of learning from the data itself seems appealing, especially due to the recent outstanding performance of (deep) feature learning methods [Sha+16; Liu+16; Wan+16; Li+17a; Ke+17; Liu+17b; Cav+17c].

Motivated by the previous considerations, we are now interested in *learning* the weights of $\varphi_{\text{kron}-\pi}$ from data. We propose to do so by taking advantage of the formal analogy between $\varphi_{\text{kron}-\pi}$ and the hidden layer of a perceptron. Since $n = 1$, we only have $\mathbf{W}^{(1)} = \mathbf{W}$ in (3.21) and we can also write

$$\varphi_{\text{kron}-\pi}(\mathbf{X}) \propto \text{tr}(\mathbf{W}^\top \mathbf{X}) = \langle \mathbf{W}, \mathbf{X} \rangle_F = \text{vec}(\mathbf{W})^\top \text{vec}(\mathbf{X}). \tag{3.68}$$

As a result, if we denote as $\mathbb{W}$ the $\nu \times d^2$ matrix which stacks by rows all the parameters $\mathbf{W} = \mathbf{W}^{(1)}$ of each independent realization of the approximated feature maps, we get that

$$\boldsymbol{\phi}_{\text{kron}-\pi}(\mathbf{X}) = \mathbb{W}\text{vec}(\mathbf{X})^\top \tag{3.69}$$

meaning that $\boldsymbol{\phi}_{\text{kron}-\pi}$ actually computes the hidden representation of a (1-layer) perceptron fed with (the vectorization of) $\mathbf{X}$ as data. Furthermore, a squeezing non-linearity (such as tanh or sigmoid) function on top of (3.69) can be actually interpreted as a sort of data normalization which is a good practice before SVM training. Since the latter can be implemented in a neural network by means of a hinge loss with weight decay, we can therefore establish a connection between our paradigm $\boldsymbol{\phi}_{\text{kron}-\pi}$ + linear SVM and a feed-forward perceptron, having one hidden layer of size ν, with sigmoid non-linearities and hinge loss with weight decay for final classification.

Let us summarize the previous findings in the following statement.

Consider $\boldsymbol{\phi}_{\text{kron}-\pi}$, set $n = 1$ and, instead of a random sampling, learn the weights $\mathbf{W} = \mathbf{W}^{(1)}$ for each $\varphi_{\text{kron}-\pi}$-component from the hidden layer of the architecture composed by a supervised feed-forward perceptron with sigmoid as non-linearities and cross entropy loss. Then, use the network to extract the feature map, that we term Log-COV-net, and use it in combination of a linear SVM. This can be interpreted as a deterministic implementation of $\varphi_{\text{kron}-e}$ and $\varphi_{\text{kron}-\pi}$ where random weights' sampling is replaced with their data-driven optimization.

3.3.1 Experiments

In this Section we will benchmark the proposed Log-COV-net against the following state-of-the art approaches in either kernel methods and feature learning paradigms.

Algorithm 4: The perceptron heuristics.

Input: A $d \times d$ input matrix $\mathbf{X}$, a training set $\boldsymbol{\mathcal{D}}$ of $d \times d$ matrices, the desired feature size ν, the probability distributions ρ over integers and $\mathcal{P}$ over real numbers, the kernel bandwith $\sigma > 0$.

Output: The ν-dim feature map $\boldsymbol{\phi}_P(\mathbf{X})$

1 Learn $\nu \times d^2$ weight matrix $\mathbb{W}$ from the hidden layer parameters of the architecture of [Cav+17c] trained on $\boldsymbol{\mathcal{D}}$.

2 **Return** $\boldsymbol{\phi}_P(\mathbf{X})$ as the multiplication of $\mathbb{W}$ by the vectorization of $\mathbf{X}$.

Kernel methods. We compare against the Fisher vectors-based encoding of [Eva+14] and the Lie group representation [Vem+14] and related Lie algebra embedding [VC16] of roto-translations. We also compare against the combination of multiple non-linear RBF kernels (Ker-RP-RBF) [Wan+15b], the sequence and dynamics compatibility kernels (SCK + DCK) [Kon+16] and Hankel matrices combined with either HMM (H-HMM) [PCSC14] or geodesic nearest neighbours method with class-protypes (H-prototypes) [Zha+16a]. Also, we consider the nearest neighbor classification performed in [JW17] through a spatio-temporal Bayesian kernel similarity. Since our approach is covariance-based, we benchmark the temporal pyramid of covariance descriptors (t-COV-pyramid) of [Hus+13], Bregman-divergence [Har+14] and the kernelized covariance operator (Ker-COV) [Cav+16]. Despite [Min+16b] applies a similar approximated-covariance paradigm, the published results only pertain to image classification. For completeness, we run the original code and applied it to 3D action recognition, denoting with rnd-logHS and QMC-AlogHS the approaches which exploit either random sampling or Quasi-Monte Carlo integration.

Feature learning approaches. We compete against the following recurrent architectures: the RNN fed on the raw joints data (J-RNN) [Sha+16] with its body part-aware variant [Du+15] and we consider Long-Short Term Memory units fed by either raw joints (J-LSTM) [Sha+16] and its improvements J-LSTM-α [Liu+16] and J-LSTM^2-α [Liu+17b], which adopt either a shallow or a deep attention module, respectively. We compare against the ensemble of deep models given by RNN-tree [Li+17b] and TSLSTM [Lee+17]
We consider the architectures proposed in [HG17b] and [Hua+17] which embed a structured input data matrix within a deep net: [HG17b] trains a deep neural network on top of covariance matrices (SPD-Net) and [Hua+17] trains on top of rotation matrices. We also compete against LieNet-3B, the 3 blocks configuration that is superior to other investigated in [Hua+17].
Also, we benchmark our approach against a few other methods which computes dynamic images (DI), image-like data structures from the joint data to encode the kinematics, and exploit them to train a convolutional neural network. Namely, we consider the J-DI$_E$-CNN [Wan+16] that exploits the Euclidean distance function between joints, J-DI$_\theta$-CNN [Li+17a] that extract DI from roto-translational representations and J-DI$_v$-CNN [Ke+17] that does the same from velocities, approximated with finite differences.

At the same time, we report the best performance obtained from Figures 3.5, 3.6 and 3.7 related to the Hadamard- [Le+13], Fourier- [RR07; Vem+10; VZ12] and Taylor-based approximations [KK12], that we indicate with H-approx, F-approx and T-approx, respectively. Ancillary, we also compare with our proposed

approximated feature maps $\boldsymbol{\Phi}_{\text{kron}-\text{e}}$ and $\boldsymbol{\Phi}_{\text{kron}-\pi}$.

The results are reported in Table 3.5, except for the comparison Log-COV-net versus [Kon+16] which is presented in Table 3.6 due to the different experimental protocol adopted from [Kon+16].

Discussion. While inspecting the performance of the approximated feature maps $\boldsymbol{\Phi}_{\text{kron}-\text{e}}$ and $\boldsymbol{\Phi}_{\text{kron}-\pi}$, in Table 3.5, we can appreciate a little improvement ($< 1\%$) over the alternative methods [RR07; Vem+10; VZ12; Le+13; KK12] in the small data regime of Florence3D, MSR-pairs, MSR-Action3D, HDM-05_14 and UTKinect (for G3D T-approx is about 1% better). This is coherent with what we found: since the aforementioned performance is mainly all achieved by $\nu = 5000$, in such a case, all methods seem to converge towards the performance of an exact kernel machine fed by (3.18). Differently, in the remaining datasets, either in the medium or big data regime, we register a significant boost in performance of $\boldsymbol{\Phi}_{\text{kron}-\text{e}}$ and $\boldsymbol{\Phi}_{\text{kron}-\pi}$ over the other approaches.

Still, we can spot how, in certain cases, $\boldsymbol{\Phi}_{\text{kron}-\text{e}}$ and $\boldsymbol{\Phi}_{\text{kron}-\pi}$ are better than methods which have been explicitly designed for action recognition: remember that, in theory, those approximations hold for any type of $d \times d$ data input. For instance, on MSR-Action3D, $\boldsymbol{\Phi}_{\text{kron}-\text{e}}$ and $\boldsymbol{\Phi}_{\text{kron}-\pi}$ improves [Hus+13] by about +15% and, on the NTU RGB+D dataset with the cross-subject protocol, the performance of [Vem+14] and [Eva+14] is improved by +4% and +16%. Eventually, on the NTU-$\times$-subject, the performance scored by $\boldsymbol{\Phi}_{\text{kron}-\text{e}}$ and $\boldsymbol{\Phi}_{\text{kron}-\pi}$ is almost on par with respect to the deep recurrent neural networks J-RNN and J-RNN-parts. Furthermore, in the middle data regime of MSRC-Kinect12 and HDM-05_{all}, $\boldsymbol{\Phi}_{\text{kron}-\text{e}}$ and $\boldsymbol{\Phi}_{\text{kron}-\pi}$ (and, in general, all the other approximated feature maps hereby considered) are scoring better than [Hus+13; Har+14; Wan+15b; Wan+16] on MSRC-Kinect12. For what concerns HDM_{all}, $\boldsymbol{\Phi}_{\text{kron}-\pi}$ is even able to beat by 5% the SoA deep learning method SPD-net [HG17b]. Such trend can be motivated by the fact that, in the middle data regime ($\sim 10^4$), the data instances are sufficiently rich to train satisfactory decision boundaries in a max margin sense, while not enough to effectively train deep models which suffer from their over-parametrization.

While moving from either $\boldsymbol{\Phi}_{\text{kron}-\text{e}}$ or $\boldsymbol{\Phi}_{\text{kron}-\pi}$ to Log-COV-net, we *always* observe a growth in performance, the latter being about +2% in the worst case and about +22% in the best one. Precisely, in the small data regime, we improved previously published state-of-the-art classification results by +0.5% on MSR-Action3D, +0.8% on MSR-pairs, +1% on HDM-05_{14} and by +2.1% on G3D.

At the same time, the gap in accuracy between Log-COV-net and $\boldsymbol{\Phi}_{\text{kron}-\text{e}}$, $\boldsymbol{\Phi}_{\text{kron}-\pi}$ grows as the size of the dataset increases: such correlation is clearly a matter of the well known fact that feature learning benefits from more data. Again, the middle data regime seems the ideal operative setting for Log-COV-net, since, to the best of our knowledge, the previously published state-of-the-art performance on MSRC-Kinect12 by +2.3% (with respect to Ker-RP-RBF [Wan+15b]) and by +10.6% on HDM-05_{all}.

On the NTU RGB+D experiments, Log-COV-net improves (by margin) Fisher vectors [Eva+14], Lie group representation [Vem+14] as well as the deep J-RNN and J-RNN-parts on the NTU-$\times$-subjects. However, when comparing with

the performance of LSTM- and CNN-based methods, Log-COV-net shows a suboptimal performance. This trend can be justified in two ways.
On the one hand, we are applying a shallow architecture with just one hidden layer while, for instance, J-DI$_\nu$-CNN and JCNN2 uses multiple deep convnets in parallel and J-LSTM^2-a conditions a deep LSTM on the output of another deep LSTM network.
On the other hand, all LSTM-based methods and J-DI$_\nu$-CNN access all the raw coordinates for each given timestamp: therefore, since we train the architecture of [Cav+17c] on covariance matrices, we can say that we are using much less data that are reduced by a factor of approximatively 1/100, being 100 the typical temporal length for the sequences on the NTU RGB+D dataset.

Despite the previous two points are a drawback in terms of classification accuracies, they results in the following operative advantages.
First, since the architecture of [Cav+17c] is shallow, there is no need for GPU acceleration neither for inference (which is nevertheless real-time), nor for the training stage (which, even on CPU, only lasts less than one hour, as opposed to one day, for instance, for the LSTM networks to be trained [Liu+17b]). Therefore, our system achieves a clear portability for deployment in real-world applications that requires real-time and scalable recognition capabilities.
Second, our representation is very compact: the experiments reported in Table 3.6, we are able to always overcome SCK and DCK in performance, even using a feature representation which is about 100 times more compact. Even on the NTU RGB+D dataset, we train the coefficients of the support vectors on top of the hidden representation of [Cav+17c] where its size is fixed to 2^8. Having only two sets of weighted elements is a very favorable operative condition as opposed to stacking several convolutional layers [Wan+16; Li+17a; Ke+17] or allocating high-dimensional tensors for back-propagating through times and train the architectures of [Du+15; Sha+16; Liu+16; Liu+17b].

This certifies in empirical terms the benefits of learning instead of sampling weights since although being a simple heuristics, the improvements in performance justifies the soundness of our proposed approach.

3.4 Conclusion

In this Chapter we propose an extended analysis of methods for action recognition which are based on temporal covariance representations.

In Section 3.1, we address the problem of covariance representations in capturing linear relationships only. Since we posit that the latter are not enough for capturing the composite nature of actions and activities, we provide a sound theoretical framework which allows to recover the kernel trick for covariance estimation. That is, by only invoking a kernel function we are able to implicitly perform a feature expansion step where linear correlations in a transformed space are equivalent to more arbitrary relationships in the original data space. Notably, we do not compute such feature expansion explicitly but, indeed, since only a kernel function needs to be evaluated, the framework allows us to compute finite kernelized covariances even when the feature transformation is infinte-dimensional - such as for the case of RBF kernels.

	Florence3D	MSR-pairs	G3D
H-approx [Le+13]	85.5	72.8	83.8
F-approx [RR07][Vem+10][VZ12]	83.4	72.2	83.4
T-approx [KK12]	84.2	73.6	**84.6**
$\Phi_{\text{kron}-\text{e}}$ *(proposed)*	***84.9***	*73.1*	*83.9*
$\Phi_{\text{kron}-\pi}$ *(proposed)*	*84.3*	***73.7***	*83.8*
rnd-LogHS [Min+16b]	88.1	79.4	87.8
QMC-logHS [Min+16b]	88.5	79.5	89.5
J-diff-DI-CNN [Ke+17]	–	90.3	–
LieNet-3B [Hua+17]	–	–	89.1
Lie Group [Vem+14]	90.7	91.4	91.1
Lie Algebra [VC16]	**91.4**	94.7	90.9
Log-COV-net (proposed)	*91.2*	***95.5***	***93.0***

	MSR-Action3D	HDM-05_{14}	UTKinect
H-approx [Le+13]	88.4	89.2	83.9
F-approx [RR07][Vem+10][VZ12]	88.2	88.6	84.0
T-approx [KK12]	89.6	88.9	84.0
$\Phi_{\text{kron}-\text{e}}$ *(proposed)*	*89.5*	*89.6*	***84.4***
$\Phi_{\text{kron}-\pi}$ *(proposed)*	***89.9***	***89.9***	*84.0*
t-COV-pyramid [Hus+13]	74.0	91.5	–
H-HMM [PCSC14]	89.0	–	86.8
rnd-logHS [Min+16b]	91.5	88.5	89.7
QMC-logHS [Min+16b]	90.6	85.4	91.3
H-prototypes [Zha+16a]	94.7	86.3	**100**
TS-LSTM [Lee+17]	–	–	97.0
J-LSTM [Liu+16]	94.8	–	97.0
Ker-RP-RBF [Wan+15b]	96.9	96.8	–
ST-BNN [JW17]	94.8	–	98.0
Ker-COV [Min+16b]	96.8	98.1	–
Log-COV-net (proposed)	***97.4***	***99.1***	*98.3*

TABLE 3.4: Classification accuracies [%] for 3D action recognition. For each table, the top part present the performance achieved by $\Phi_{\text{kron}-\pi}$ and $\Phi_{\text{kron}-\text{e}}$ against other alternative approximating schemes [RR07; Vem+10; VZ12; Le+13; KK12]: within this class of methods, the best accuracy is highlighted in bold. At the same time, in the bottom part of each table, Log-COV-net is compared against state-of-the-art approaches and, among them, the best performance is marked by bold and underlined. All the performance achieved by methods proposed in this Chapter ($\Phi_{\text{kron}-\text{e}}$, $\Phi_{\text{kron}-\pi}$ and Log-COV-net) are in italic.

	MSRC-Kinect12	HDM-05_{all}
H-approx [Le+13]	92.4	63.7
F-approx [RR07][Vem+10][VZ12]	92.2	64.0
T-approx [KK12]	92.8	62.0
$\Phi_{\text{kron}-\text{e}}$ *(proposed)*	*92.3*	*65.0*
$\Phi_{\text{kron}-\pi}$ *(proposed)*	***95.6***	***66.5***
t-COV-pyramid [Hus+13]	89.2	–
Bregman-div [Har+14]	89.9	58.2
Ker-RP-RBF [Wan+15b]	92.3	66.2
J-DI_E-CNN [Wan+16]	93.1	–
Ker-COV [Cav+16]	95.0	–
rnd-logHS [Min+16b]	97.1	58.1
QMC-logHS [Min+16b]	96.2	60.2
SPD-net [HG17b]	–	61.4
Log-COV-net (proposed)	***98.5***	***72.0***

	NTU-×-subject	NTU-×-view
H-approx [Le+13]	51.5	50.6
F-approx [RR07][Vem+10][VZ12]	50.7	50.6
T-approx [KK12]	50.8	51.0
$\Phi_{\text{kron}-\text{e}}$ *(proposed)*	*50.7*	*49.4*
$\Phi_{\text{kron}-\pi}$ *(proposed)*	***54.0***	***54.1***
Fisher Vectors [Eva+14]	38.6	41.4
Lie Group [Vem+14]	50.1	52.8
J-RNN [Sha+16]	56.3	64.0
J-RNN-parts [Du+15]	59.1	64.1
LieNet-3B [Hua+17]	61.4	67.0
J-LSTM [Sha+16]	60.7	67.3
J-LSTM-α [Liu+16]	69.2	77.7
J-DI_E-CNN [Wan+16]	73.4	75.2
J-$LSTM^2$-α [Liu+17b]	74.4	82.8
TS-LSTM [Lee+17]	74.6	81.3
RNN-tree [Li+17b]	74.6	83.2
J-DI_θ-CNN [Li+17a]	76.2	82.3
J-DI_ν-CNN [Ke+17]	**79.6**	**84.8**
Log-COV-net (proposed)	*60.9*	*63.4*

TABLE 3.5: Classification accuracies [%] for 3D action recognition. For each table, the top part present the performance achieved by $\Phi_{\text{kron}-\pi}$ and $\Phi_{\text{kron}-\text{e}}$ against other alternative approximating schemes [RR07; Vem+10; VZ12; Le+13; KK12]: within this class of methods, the best accuracy is highlighted in bold. At the same time, in the bottom part of each table, Log-COV-net is compared against state-of-the-art approaches and, among them, the best performance is marked by bold and underlined. All the performance achieved by methods proposed in this Chapter ($\Phi_{\text{kron}-\text{e}}$, $\Phi_{\text{kron}-\pi}$ and Log-COV-net) are in italic.

	Florence3D⋆	UTKinect	MSR-Action3D⋆	MSR-Action3D
SCK [Kon+16]	92.98	96.1	90.72	93.5
DCK [Kon+16]	93.03	97.5	86.30	91.7
SCK+DCK [Kon+16]	95.23	98.2	91.45	94.0
Log-COV-net (proposed)	***97.25***	***98.3***	***96.30***	***97.4***

TABLE 3.6: Classification accuracies [%] of Log-COV-net against [Kon+16]. Best results are bold and underlined, the symbol ⋆ indicates that we used the alternative training/testing split adopted in [Kon+16].

In Section 3.2 we still focus on RBF kernels and, in particular, we tackle the scalability issue which arises from the fact that, in order to train an RBF kernel machines, Gram matrices need to be computed and this is simply not affordable in a big data regime due to the overall quadratic complexity. As a remedy, we propose an explicit and approximated feature map with randomly sampled weights. Once averaging upon all their realizations, we are able to devise a reliable estimator of the effective kernel such that the proposed approximation has no bias and, moreover, as the dimensionality ν of the feature representation grows, the variance decreases to zero as $\frac{1}{\nu^3}$. All such favorable theoretical properties opens up to an efficient pipeline to train a linear machine - which is actually not a problem even in the big data regime - on top of the proposed feature representation: our theory can guarantee that all such approach is indeed a scalable surrogate for an exact kernel machine. Moreover, empirical evidence shows a minimal drop in performance between the two approaches, even when one selects $\nu = 10, 20, 50$, therefore promoting for compact and effective approximated feature representations.

Finally, in Section 3.3, we pursue the idea that, although capturing all temporal correlations between joints leads to a complete overview on the actions' kinematics, nevertheless, it can be the case that some of those correlations are more important than others. In order to spot the latter ones, we proposed a data-driven approach to re-weight covariance representations in order that we discriminatively train one hidden-layered network to magnify the most relevant correlations which enable to better disambiguate between actions. Our approach allows us to validate the claim that, since we posit that actions' kinematics is proficiently captured by covariance representations, there is no need for the architecture to be deep in order to score a favorable performance.

In all cases, the proposed methods are evaluated against state-of-the-art competitors for actual recognitions of human actions from 3D skeletal joints. In all cases, effectiveness of the proposed techniques is evaluated in terms of improvements over previously score state-of-the-art classification performance, ultimately assessing the reliability of the investigated methods.

Chapter 4

Huber Loss Regression: Robust correlation-based learning from multiple views

Multi-view data have become increasingly available in real-world applications where examples are described by different feature sets or different "views", such as image & text, audio & video, and web pages & click-through data (see Fig. 4.1). Taking into account such wellness of data modalities, designing methods which can only leverage on one data representation at a time seems a clear limitation. On the other hand, however, combining different views which relate to the same data point is not straightforward at all, since a direct concatenation of different feature vectors is problematic under several point of views. Indeed, since one needs to re-normalize the concatenated vector, if some feature components are much lower in magnitude than others, and, while doing so, their effect in learning the model may be cut out. Additionally, when multiple encodings are juxtaposed, the resulting representation is extremely high-dimensional and, in the case of a reduced number of training example, the so called curse of dimensionality problem [Bis06] is detrimental for the model's optimization.

In contrast to single-view paradigms, multi-view learning separately introduces one embedding per view, attempting to solve the recognition task in each of this embedding space separately and ultimately fusing the contribution coming from each of the subspaces. To be concrete, in the case of a classification problems from multiple views, intra-class boundaries are learnt separately by means of each view and then are combined together by promoting mutual agreement in between. And, usually, the whole pipeline is implemented as a whole optimization problem which jointly learns, first, how to solve the recognition problem in each single view embedding and, second, how to combine the views.

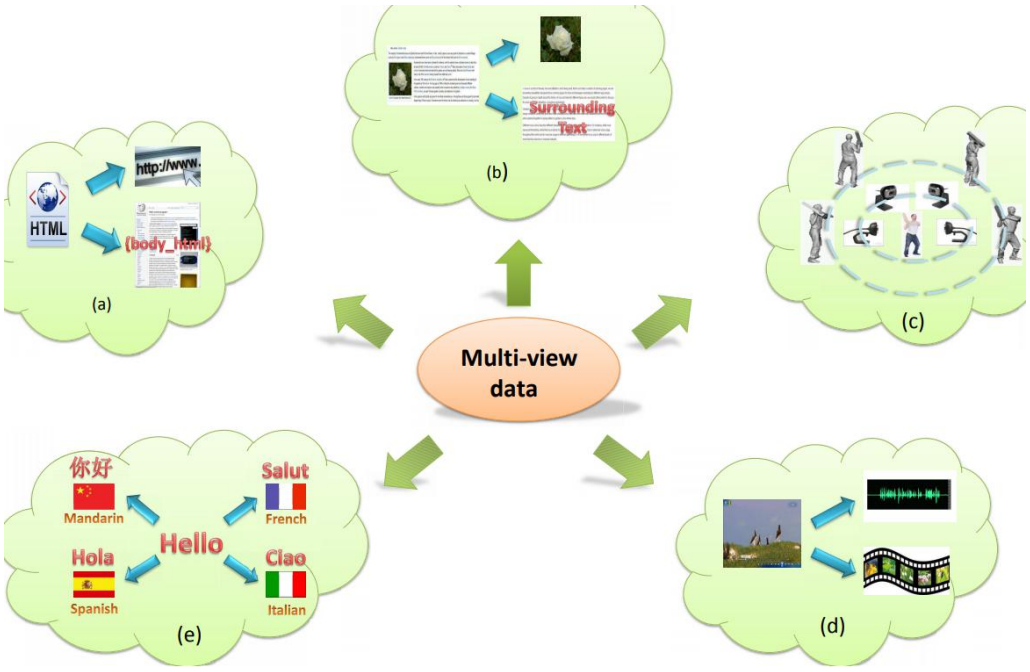


FIGURE 4.1: Multi-view data: a) a web document can be represented by its url and words on the page, b) a web image can be depicted by its surrounding text separate to the visual information, c) images of a 3D object taken from different viewpoints, d) video clips are combinations of audio signals and visual frames, e) multilingual documents have one view in each language. Image and caption courtesy of [Sun13]

In order to make this paradigm effective in practice, however, one need to require some classical, statistical assumptions between the views [Sun13].

1. One has to assume that each single view separately is able to provide cues for the recognition problem to be solved.
2. The views correlate together in the sense that the information they provide is complementary.
3. In either supervised or semi-supervised settings, each view is conditionally independent to any other one given the class label.

The first assumption is rather free in an usual applicative domain and can be accommodated by a principled choice of which feature representation has to be used and, in general, even if some of the views are less rich in information than others, still multi-view learning may be proficiently applied [Sun13].

For the second assumption, many approaches in multi-view learning have tried to explicitly write down optimization as to promote correlation between different views, so that the learnt representation benefits from all. In fact, as one of the most famous and earliest approaches proposed within multi-view learning literature, Hotelling [Hot36] seeks to learn a linear p-dimensional subspace where two different n- and n'-dimensional spaces of views need to be projected on. To do so, optimization writes as

$$\max_{W \in \mathbb{R}^{p \times d}, W' \in \mathbb{R}^{p \times d'}} \sum_{i} \operatorname{cov}(W s_i, W' s'_i) \tag{4.1}$$

where, given samples s_i from the first view and s'_i from the second, two linear transformations, W and W' respectively, are learnt so that the transformed data show maximal agreement in the sense of the covariance function (2.1). That is, the problem (4.1) impose that the learnt projection falls in a p-dimensional subspace in which the projections are well closed to each other in the sense that they shows a maximal amount of linear correlation in between them. Such method is termed Canonical Correlation Analysis (CCA) has been applied to a variety of visual recognition tasks, being also extended to accommodate for more elaborated notions of alignment rather than linear correlation (for a comprehensive review, see [Hai+, Chapter 8]).

Critically, the last assumption, which requires each view to be independent from the others once conditioned on the class label, is the most hard one to be satisfied in either theory or practice. There are some theoretical papers which have provided a few insights on this topic. For instance, [BM98] proved that having two views conditionally independent given the class label is a sufficient condition to guarantee that the learning task can be proficiently performed, in terms of a controlled generalization error even in the case of views which are affected by noise. Similarly, [Das+01] provides a probabilistic approximated condition to ensure that the conditional independence assumption can be relaxed with alternative requirements which are easier to check.

Nevertheless, to the best of our knowledge, there is a lack of multi-view learning methods which are explicitly designed to work in conditions which, hypothetically, the latter assumption is not satisfied. In this chapter, we deliberately focus on this problem and we consider the case when the annotations which are used in a (semi-)supervised learning framework, may be corrupted by noise. In such a case, indeed, due to the ambiguity in the annotation, it may be the case that the conditionally independence assumption is no more valid. In addition, even if we assume the ideal case where the set of annotations is only marginally affected by noise, therefore having a part of clean labels against the remaining ones which are noisy, we actually face the situation when the assumption of the conditional independence is valid only for some examples (the ones which own a clean annotation). Therefore, for a method to operate in such challenging condition, we should allow the possibility of automatically scan the annotations/labels, spot which one are affected by noise and ultimately remove the corresponding instances that are recognized as outliers.

Due to the fact that semi-supervised approaches [Wan+12a; Hua+14; Don+16; Den+14; Tri+15] play a substantial role to support the learning stage by exploiting unlabeled examples since annotations are usually provided by human operators, they are frequently prone to errors and noisy in general, making them rather misleading. Hence, it is of utmost importance to devise algorithms which are able to automatically analyze the data as to guarantee robustness towards outliers. In the literature, several works have tackled such a problem [Hua+12; Don+16] and since the archetypal work [Hub64], many robust regression and classification frameworks [MM00; AZ05; LLZ11; Kha+13] successfully leveraged on the Huber loss function. Denoted in this work by H_ξ, the Huber loss is defined as

$$H_\xi(y) = \begin{cases} \frac{y^2}{2} & \text{if } |y| \leq \xi \\ \xi|y| - \frac{\xi^2}{2} & \text{otherwise,} \end{cases} \tag{4.2}$$

where $\xi > 0$ and $y \in \mathbb{R}$. The Huber loss as in (4.2) generalizes both the quadratic loss and the absolute value, which can be recovered in the extremal cases $\xi \to +\infty$ and $\xi \to 0$, respectively. H_ξ has been shown to be robust against outliers [Hub64] since, in a neighborhood of the origin, it penalizes small errors in a more smoother way than absolute value, whereas, when $|y| \geq \xi$, the linear growth plays an intermediate role between over-penalization (quadratic loss) and under-penalization (absolute value) of larger errors. Globally, it resumes the positive aspect of the two losses while remarkably mitigating their weaknesses. However, as major drawback of H_ξ, there is no closed-form solution to optimize it and, as a consequence, iterative schemes (such as quadratic programming [AZ05] or self-dual minimization [LLZ11]) were previously exploited for either the original Huber loss [MM00; LLZ11] or its spurious versions (hinge-Huber [AZ05] or the huberized Laplacian [Kha+13]). Moreover, in all cases, additional computational efforts have to be spent in order to fix the threshold ξ, such as statistical efficiency analysis [MM00].

In this work we face all the aforementioned issues trough the following main contributions.

1. We derive a novel theoretical solution to exactly optimize the Huber loss in a general multi-view and manifold regularization setting [Min+13], in order to guarantee a broad applicability of the developed formalism. In such a multi-view learning framework, we can adopt a different kernel function on the basis of the type of data we are analyzing (Fig. 4.1) so that we can perform a principled low level encoding, by jointly learning a decision function in a multi-modal approach. Such approach make the method applicable to potentially any type of input data.

2. We devise the novel Huber Loss Regression (HLR) algorithm to efficiently implement the proposed solution and avoid classical iterative schemes [AZ05; LLZ11], with two additional characteristics as quoted in the following.
   Auto-unlabeling. While taking advantage of both labeled and unlabeled training samples, the former ones are inspected so that HLR automatically removes those annotations violating a specific numerical check, whenever recognized as either noisy or inconsistent for learning stage.
   Adaptive threshold. Unlike [MM00; AZ05; LLZ11; Kha+13], HLR automatically learns ξ in a data-driven fashion without increasing the computational complexity of the whole pipeline.

3. Throughout an extensive empirical evaluation, we validate the proposed technique, which allows to score competitive results in curve fitting, learning with noisy labels, classical regression problems and crowd counting applications.
   While using variegate types of data and addressing diverse problems, HLR is able to outperform state-of-the-art regression algorithms.

Precisely, the rest of the chapter is organized as follows.

4.1 Background and problem formulation

Let $y \in \mathcal{Y} \subseteq \mathbb{R}$ a scalar target variable and $x \in \mathcal{X} \subseteq \mathbb{R}^d$ an independent input; practically, x will encode a feature vector. Inside the space of functions $\mathcal{H}$, our goal is finding the hypothesis $h\colon \mathcal{X} \to \mathcal{Y}$ whose optimality as regression function is measured by means of the empirical risk $\frac{1}{\ell}\sum_i V(y_i - h(x_i))$, where $V\colon \mathcal{Y} \to [0, +\infty)$, non-negative, is called the *loss function*. $V(y_i - h(x_i))$ measures how good is $h(x_i)$ in predicting y_i : one example is given by the quadratic loss $(y_i - h(x_i))^2$. In order to learn h, a training set $\{(x_1, y_1), \dots, (x_\ell, y_\ell)\}$ is necessary, where the instance x_i is paired with corresponding response values y_i. Precisely, h is optimization-based selected: h minimizes the regularized empirical risk $\frac{1}{\ell}\sum_i V(y_i - h(x_i)) + \lambda\|h\|^2$, where $\lambda > 0$ is a Lagrange multiplier for the Tichonov regularization term and the $\|\cdot\|$ is a functional norm, defined over a suitable space $\mathcal{H}$, aiming at controlling the complexity of $h \in \mathcal{H}$. In the following paragraphs, we will select the hypothesis space $\mathcal{H}$ and formulate the manifold regularization regression problem.

Hypothesis space. For any $\alpha = 1, \dots, m$, let $\kappa^\alpha\colon \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ be a Mercer kernel [SS02], that is a symmetric and positive semi-definite function. Let us define $K(x, x') = \operatorname{diag}(\kappa^1(x, x'), \dots, \kappa^m(x, x')) \in \mathbb{R}^{m\times m}$, where $x, x' \in \mathcal{X}$. Consider $\mathcal{S}_0$ the space of functions $f(x) = \sum_{i=1}^n K(x, x_i)a_i$ with $x_1, \dots, x_n \in \mathcal{X}$ and $a_1, \dots, a_n \in \mathbb{R}^m$. Define the norm $\|f\|_K^2 = \sum_{i,j=1}^n a_i K(x_i, x_j)a_j$. The reproducing kernel Hilbert space $\mathcal{S}_K$ (related to K) is the completion of $\mathcal{S}_0$, adding all the limits of converging Cauchy sequences [Car+06]. The final hypothesis space $\mathcal{H}$ is the image of $\mathcal{S}_K$ through the map $c^\top\colon \mathcal{S}_K \to \mathcal{H}$ defined as $h = c^\top f$ for some $c = [c^1, \dots, c^m] \in \mathbb{R}^m$. Thus, for any $x \in \mathcal{X}$, denoting a_i^α the α-th component of a_i, we design our hypothesis $h(x)$ as

$$c^\top \left(\sum_{i=1}^n K(x, x_i)a_i \right) = \sum_{i=1}^n \sum_{\alpha=1}^m c^\alpha \kappa^\alpha(x, x_i)a_i^\alpha. \tag{4.3}$$

Kernel matrix K produces the high-level descriptor $\sum_i K(x, x_i)a_i \in \mathbb{R}^m$ which allows to encode separately some intrinsic heterogeneities of independent variables. This is a very appealing perspective in multivariate regression; moreover, exploiting mutual differences between features has been shown to be effective (see [Bia+13]). Finally, this formulation is similar to the concept of *view* in [Min+13], with the crucial difference that each view encodes separately each feature.

Manifold regularization regression. Consider a training set $\mathbf{z}$ made of ℓ pairs $(x_1, y_1), \dots, (x_\ell, y_\ell) \in \mathcal{X} \times \mathcal{Y}$ and u additional inputs $x_{\ell+1}, \dots, x_{\ell+u} \in \mathcal{X}$. This formulation is very general: $u = 0$ leads to a fully supervised setting in which all $x_i \in \mathcal{X}$ are provided with targets $y_i \in \mathcal{Y}$. If $u > 0$, we exploit $x_{\ell+1}, \dots, x_{\ell+u}$ to infer the geometrical information of $\mathcal{X}$. Learning unsupervised concepts is both familiar to human brain and useful to improve algorithms generalization [Bel+06]. Now, consider

$$\mathcal{J}_{\lambda,\gamma}(f) = \frac{1}{\ell}\sum_{i=1}^{\ell} V(y_i - c^\top f(x_i)) + \lambda\|f\|_K^2 + \gamma\|f\|_M^2. \tag{4.4}$$

It consists in three terms. $\frac{1}{\ell}\sum_{i=1}^{\ell} V(y_i - c^\top f(x_i))$ is the aboved mentioned empirical risk; $\|f\|_K^2$ is the Tichonov regularizer which controls the model complexity and it is scaled by $\lambda > 0$. Parameter $\gamma \geq 0$ weighs the manifold regularizer which infer the geometrical information of the feature space [Bel+06]. Indeed, $\|f\|_M^2 = \sum_{i,j=1}^{u+\ell} f(x_i)^\top M_{ij} f(x_j)$ captures the mutual position of instances $x_1, \ldots, x_{u+\ell} \in \mathcal{X}$. Analytically, $\|\cdot\|_M$ is the norm induced by the symmetric and positive definite matrix $M \in \mathbb{R}^{m\times m}$.

Apparently, optimizing $J_{\lambda,\gamma}(f)$ over the Reproducing Kernel Hilbert Space related to K seems burdensome, due to the infinite dimension of $\mathcal{S}_K$. In addition, like all the classical optimization problems, some efforts must be spent to provide existence and uniqueness for the optimizer $f^\star$ such that $f^\star = \min_{f\in\mathcal{S}_K} J_{\lambda,\gamma}(f)$. Thanks to Representer Theorem [Min+13], all these issues are solved: $f^\star$ exists, is unique and it is furthermore computable using the expansion

$$f^\star(x) = \sum_{j=1}^{u+\ell} K(x, x_j) w_j, \tag{4.5}$$

in terms of some $w_1, \ldots, w_{u+\ell} \in \mathbb{R}^m$. Since coefficients $\mathbf{w} = [w_1, \ldots, w_{u+\ell}]$ define explicitly $f^\star$, we are able to optimize over the $m(u+\ell)$-dimensional variable $\mathbf{w}$, instead of over $\mathcal{S}_K$. Taking advantage of such theoretical results, we are able to obtain the following result.

Theorem 5 (General closed-form solution). *The coefficients $w_1^\star, \ldots, w_{u+\ell}^\star \in \mathbb{R}^m$ which defines as in (4.5) the minimizer $f^\star$ of (4.4) are computable as the solution of the following optimization problem.*

$$V'\left(y_i - \sum_{j=1}^{u+\ell} c^\top K(x_i, x_j) w_j\right) c = 2\ell\lambda w_i + 2\ell\gamma \sum_{j,h=1}^{u+\ell} M_{ij} K(x_j, x_h) w_h \tag{4.6}$$

for $i = 1, \ldots, \ell$; and, when $i = \ell+1, \ldots, u+\ell$,

$$\lambda w_i + \gamma \sum_{j,h=1}^{u+\ell} M_{ij} K(x_j, x_h) w_h = 0. \tag{4.7}$$

Proof. Preliminary, let's assume that the loss function V is differentiable: we will relax such assumption later. Implement the representation (4.5) inside the manifold regularization framework (4.4). Then, one gets the following equivalent minimization problem

$$\mathcal{J}_{\lambda,\gamma}(\mathbf{w}) = \frac{1}{\ell}\sum_{i=1}^{\ell} V\left(y_i - \sum_{j=1}^{u+\ell} c^\top K(x_i, x_j) w_j\right) + \lambda \sum_{j,k=1}^{u+\ell} w_j^\top K(x_j, x_k) w_k + \\ + \gamma \sum_{i,j=1}^{u+\ell} \sum_{h,k=1}^{u+\ell} w_h^\top K(x_h, x_i) M_{ij} K(x_j, x_k) w_j. \tag{4.8}$$

The optimization domain is $\mathbb{R}^{m(u+\ell)}$, since $\mathbf{w}$ collect $w_1, \ldots, w_{u+\ell}$ and, for any $p = 1, \ldots, u+\ell$, we have $w_p = [w_p^1, \ldots, w_p^m]^\top$. For each λ, γ, the minimizer of (4.8) is also a stationary point for $\nabla\mathcal{J}_{\lambda,\gamma}$. Thus, let's compute $\dfrac{\partial\mathcal{J}_{\lambda,\gamma}}{\partial w_p^\eta}$, the derivative of $\mathcal{J}_{\lambda,\gamma}$ with respect to w_p^η for any $p = 1, \ldots, u+\ell$ and $\eta = 1, \ldots, m$.

We have

$$\frac{\partial \mathcal{J}_{\lambda,\gamma}}{\partial w_p^\eta} = -\frac{1}{\ell}\sum_{i=1}^{\ell} V'\left(y_i - \sum_{j=1}^{u+\ell} c^\top K(x_i,x_j)w_j\right)\sum_{h=1}^{u+\ell}\sum_{\alpha=1}^{m} c^\alpha \kappa^\alpha(x_i,x_h)\frac{\partial w_h^\alpha}{\partial w_p^\eta} + \\ + \lambda\sum_{j,k=1}^{u+\ell}\sum_{\alpha=1}^{m}\frac{\partial w_j^\alpha}{\partial w_p^\eta}\kappa^\alpha(x_j,x_k)w_k^\alpha + \lambda\sum_{j,k=1}^{u+\ell}\sum_{\alpha=1}^{m} w_j^\alpha\kappa^\alpha(x_j,x_k)\frac{\partial w_k^\alpha}{\partial w_p^\eta} + \\ + \gamma\sum_{i,j=1}^{u+\ell}\sum_{h,k=1}^{u+\ell}\sum_{\alpha,\beta=1}^{m}\frac{\partial w_h^\alpha}{\partial w_p^\eta}\kappa^\alpha(x_i,x_h)M_{ij}^{\alpha\beta}\kappa^\beta(x_j,x_k)w_k^\beta + \\ + \gamma\sum_{i,j=1}^{u+\ell}\sum_{h,k=1}^{u+\ell}\sum_{\alpha,\beta=1}^{m}\kappa^\alpha(x_i,x_h)w_h^\alpha M_{ij}^{\alpha\beta}\kappa^\beta(x_j,x_k)\frac{\partial w_k^\beta}{\partial w_p^\eta}, \tag{4.9}$$

In order to simplify (4.9), it is useful to introduce Delta Dirac δ_{mn} defined as $\delta_{mn} = 1$ when $m = n$ and $\delta_{mn} = 0$ otherwise. Moreover, we exploit the relationship

$$\frac{\partial w_i^\alpha}{\partial w_j^\beta} = \delta_{\alpha\beta}\delta_{ij},$$

valid for any $\alpha,\beta = 1,\dots,m$ and $i,j = 1,\dots,u+\ell$. Thus,

$$\frac{\partial \mathcal{J}_{\lambda,\gamma}}{\partial w_p^\eta} = -\frac{1}{\ell}\sum_{i=1}^{\ell} V'\left(y_i - \sum_{j=1}^{u+\ell} c^\top K(x_i,x_j)w_j\right)\sum_{h=1}^{u+\ell}\sum_{\alpha=1}^{m} c^\alpha \kappa^\alpha(x_i,x_h)\delta_{\alpha\eta}\delta_{hp} + \\ + \lambda\sum_{j,k=1}^{u+\ell}\sum_{\alpha=1}^{m}\delta_{\alpha\eta}\delta_{jp}\kappa^\alpha(x_j,x_k)w_k^\alpha + \lambda\sum_{j,k=1}^{u+\ell}\sum_{\alpha=1}^{m} w_j^\alpha\kappa^\alpha(x_j,x_k)\delta_{\alpha\eta}\delta_{pk} + \\ + \gamma\sum_{i,j=1}^{u+\ell}\sum_{h,k=1}^{u+\ell}\sum_{\alpha,\beta=1}^{m}\delta_{\alpha\eta}\delta_{hp}\kappa^\alpha(x_i,x_h)M_{ij}^{\alpha\beta}\kappa^\beta(x_j,x_k)w_k^\beta + \\ + \gamma\sum_{i,j=1}^{u+\ell}\sum_{h,k=1}^{u+\ell}\sum_{\alpha,\beta=1}^{m}\kappa^\alpha(x_i,x_h)w_h^\alpha M_{ij}^{\alpha\beta}\kappa^\beta(x_j,x_k)\delta_{\beta\eta}\delta_{pk},$$

and

$$\frac{\partial \mathcal{J}_{\lambda,\gamma}}{\partial w_p^\eta} = -\frac{1}{\ell}\sum_{i=1}^{\ell} V'\left(y_i - \sum_{j=1}^{u+\ell} c^\top K(x_i,x_j)w_j\right) c^\eta\kappa^\eta(x_i,x_p) + \\ + \lambda\sum_{k=1}^{u+\ell} w_k^\eta\kappa^\eta(x_p,x_k) + \lambda\sum_{j=1}^{u+\ell} w_j^\eta\kappa^\eta(x_j,x_p) + \\ + \gamma\sum_{i,j=1}^{u+\ell}\sum_{k=1}^{u+\ell}\sum_{\beta=1}^{m}\kappa^\eta(x_i,x_p)M_{ij}^{\eta\beta}\kappa^\beta(x_j,x_k)w_k^\beta + \\ + \gamma\sum_{i,j=1}^{u+\ell}\sum_{h=1}^{u+\ell}\sum_{\alpha=1}^{m}\kappa^\alpha(x_i,x_h)w_h^\alpha M_{ij}^{\alpha\eta}\kappa^\eta(x_j,x_p). \tag{4.10}$$

To rearrange equation (4.10), remember the functional symmetry of Mercer

kernels $\kappa^1, \dots, \kappa^m$ and $M_{ij}^{\alpha\beta}$. In formulæ, we have $K(x, x') = K(x', x)$, that is, for any $x, x' \in \mathbb{R}^d$, $\kappa^\alpha(x, x') = \kappa^\alpha(x', x)$ for every $\alpha = 1, \dots, m$. Also, $M_{ij}^{\alpha\beta} = M_{ij}^{\beta\alpha} = M_{ji}^{\alpha\beta} = M_{ji}^{\beta\alpha}$ for each $i, j = 1, \dots, u + \ell$ and $\alpha, \beta = 1, \dots, m$. Then, one sees

$$\frac{\partial \mathcal{J}_{\lambda,\gamma}}{\partial w_p^\eta} = -\frac{1}{\ell} \sum_{i=1}^{\ell} V' \left(y_i - \sum_{j=1}^{u+\ell} c^\top (K(x_i, x_j) w_j) \right) c^\eta \kappa^\eta(x_i, x_p) + 2\lambda \sum_{k=1}^{u+\ell} w_k^\eta \kappa^\eta(x_p, x_k) + \\ + 2\gamma \sum_{i,j=1}^{u+\ell} \sum_{k=1}^{u+\ell} \sum_{\beta=1}^{m} \kappa^\eta(x_i, x_p) M_{ij}^{\eta\beta} \kappa^\beta(x_j, x_k) w_k^\beta.$$

After vectorizing with respect to $\eta = 1, \dots, m$, the derivative $\frac{\partial \mathcal{J}_{\lambda,\gamma}}{\partial w_p}$ equals to

$$-\frac{1}{\ell} \sum_{i=1}^{\ell} V' \left(y_i - \sum_{j=1}^{u+\ell} c^\top (K(x_i, x_j) w_j) \right) K(x_p, x_i) c + \\ + 2\lambda \sum_{k=1}^{u+\ell} K(x_p, x_k) \omega_k + 2\gamma \sum_{i,j=1}^{u+\ell} \sum_{k=1}^{u+\ell} K(x_p, x_i) M_{ij} K(x_j, x_k) w_k. \tag{4.11}$$

Expression (4.11) rewrites $\sum_{i=1}^{u+\ell} K(x_p, x_i) \psi_i$, once defined, for any $i = 1, \dots, u + \ell$,

$$\psi_i = -\mathbf{I}(i \le \ell) \frac{1}{\ell} V' \left(y_i - \sum_{j=1}^{u+\ell} c^\top K(x_i, x_j) w_j \right) c + 2\lambda \omega_i + 2\gamma M_{ij} K(x_j, x_k) w_k.$$

If we set $\psi_1 = \cdots = \psi_{u+\ell} = 0$, then, from equation (4.28), $\frac{\partial \mathcal{J}_{\lambda,\gamma}}{\partial w_1} = \cdots = \frac{\partial \mathcal{J}_{\lambda,\gamma}}{\partial w_{u+\ell}} = 0$ and this will lead to a solution of our system. But, this is the only solution we have since, as argumented, the optimization problem (4.23) has unique solution. Globally,

$$2\ell\lambda w_i + 2\ell\gamma \sum_{j,h=1}^{u+\ell} M_{ij} K(x_j, x_h) w_h = V' \left(y_i - \sum_{j=1}^{u+\ell} c^\top K(x_i, x_j) w_j \right) c \tag{4.12}$$

for $i = 1, \dots, \ell$; and, when $i = \ell + 1, \dots, u + \ell$,

$$2\lambda w_i + 2\gamma \sum_{j,h=1}^{u+\ell} M_{ij} K(x_j, x_h) w_h = 0. \tag{4.13}$$

Equations (4.12) and (4.13) are exactly the thesis, providing the closed form solution for our minimization problem (4.6)–(4.7).

Let's now relax the assumption of differentiability for the loss function. The only hypothesis that must be really considered is the non-negativity of the loss. Indeed, from such requirement the convexity of V is straightforward, thus interpreting V' as the sub-derivative, which share the same formal properties

of the analytic derivative (see [Rud66]). For the sake of completeness, let's us briefly sketch how to deduce the convexity of the map V. As any space of linear combination of functions, $\mathcal{S}_0$ is a convex set [BV04] and the completion preserves convexity [Rud66]. Thus, any $f \in \mathcal{S}_K$ is a convex function, and the composition $y_i - c^\top f(x_i)$ is straightforwardly convex since all the combined functions are convex. Finally, $V(y_i - c^\top f(x_i))$ is convex, being the composition of a convex map with a non-negative one [BV04]. □

Formulas (4.6) and (4.7) provide a closed-form solution for (4.4) valid with respect of any loss V under the assumption that V is (sub)differentiable. It is a remarkable result since, in general, equations (4.6) and (4.7) are valid for any loss, while all the classical methods are loss-specific. For example, normal equations are used for quadratic loss in regularized least square regression [SL12].

However, the main difference is that, in order to make (4.6) and (4.7) computable, one need to specify V and, also, the possibility of explicitly computing the derivative of V is obligatorily required. In order to make an example, let us consider the following case.

Multi-view Learning with quadratic loss [Min+13]. In (4.6), fix the loss function V to be the quadratic one, that is, $V(y) = y^2$. Thus, the problem rewrites in the following linear system.

$$\ell\lambda w_i + \ell\gamma \sum_{j,h=1}^{u+\ell} M_{ij} K(x_j, x_h) w_h = \begin{cases} \left(y_i - \sum_{j=1}^{u+\ell} c^\top K(x_i, x_j) w_j \right) c & \text{if } i = 1, \dots, \ell \\ 0 & \text{otherwise.} \end{cases} \tag{4.14}$$

After little changes in notation, (4.14) turns exactly into Proposition 1 of [Min+13], where a similar objective functional was optimized, once the loss function was fixed to be quadratic. Thus, we have generalized the scalar multi-view learning framework [Min+13], which can be recovered as a particular case of our general closed form solution provided by (4.6) and (4.7).

4.2 Robust multi-view learning with the Huber Loss

Although many multi-view learning paradigms have been proposed throughout the previous years [Sun13], and, jointly, several robust regression frameworks have been envisaged [RL05], the problem of validating multi-view learning problems in the case of noisy annotations has been addressed only in theoretical terms so far [BM98; Das+01]. Therefore, to the best of our knowledge, in the present chapter of this thesis, we firstly propose a method which learns by correlating multiple views which refers to data whose annotations are assumed to be corrupted by noise. In such a case, the prediction of y may be corrupted during the combination of the view-specific predictors $f^1(x^1), \dots, f^m(x^m)$ due the propagation of the noise from one view x^j. Alternatively, annotations may be affected by noise and also this plays a part.

To cope with the latter issue, in this section, we adopt a semi-supervised framework, where $f = [f^1, \dots, f^m]$ are learnt from a training set $\mathbf{D}$, composed by ℓ labeled instances $(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_\ell, y_\ell) \in \mathcal{X} \times \mathbb{R}$ and u additional unlabeled inputs $\mathbf{x}_{\ell+1}, \dots, \mathbf{x}_{\ell+u} \in \mathcal{X}$. In addition, as a proxy to promote robustness, we propose to consider the optimization problem which arises by the minimization of the objective functional

$$J_{\lambda,\gamma}(f) = \frac{1}{\ell} \sum_{i=1}^{\ell} H_\xi(y_i - c^\top f(\mathbf{x}_i)) + \lambda \|f\|_K^2 + \gamma \|f\|_M^2, \tag{4.15}$$

where $\frac{1}{\ell} \sum_{i=1}^{\ell} H_\xi(y_i - c^\top f(\mathbf{x}_i))$ represents the empirical risk and is defined by means of the Huber loss (4.2). It measures how well the ground truth output y_i is predicted through the linear combination $c^\top f(\mathbf{x}_i)$ of views $f^1(\mathbf{x}_i^1), \dots, f^m(\mathbf{x}_i^m)$, being $c^1, \dots, c^m > 0$. In (4.4), for any $\lambda, \gamma \geq 0$, the norms $\|\cdot\|_K$ and $\|\cdot\|_M$ regularize the solution f.

Specifically, the term $\|f\|_K^2$ is a Tichonov regularizer which controls the complexity of the solution, avoiding both under- and over-fitting. As the space of functions to search for f, we consider the Reproducing Kernel Hilbert Space (RKHS) related to $K(\mathbf{x}, \mathbf{z}) = \operatorname{diag}(\kappa^1(x^1, z^1), \dots, \kappa^m(x^m, z^m))$, being, for any $\alpha = 1, \dots, m$, $\kappa^\alpha \colon \mathcal{X}^\alpha \times \mathcal{X}^\alpha \to \mathbb{R}$ a Mercer kernel [SS02], that is a symmetric and positive semi-definite function. The shape of a diagonal kernel is finalized to encode each view separately. Also, considering the RKHS is advantageous for its formal properties [SS02]. Precisely, the optimization problem (4.4) is provided by existence and uniqueness of the minimizer $f^\star = \arg\min_{f \in \mathcal{S}_k} J_{\lambda,\gamma}(f)$, which can be expanded as

$$f^\star(\mathbf{z}) = \sum_{j=1}^{u+\ell} K(\mathbf{z}, \mathbf{x}_j) w_j, \tag{4.16}$$

for $w_1, \dots, w_{u+\ell} \in \mathbb{R}^m$ [SS02]. Also, the norm $\|\cdot\|_K$ is easily computable for $f^\star$ as in (4.16) through the following expression.

$$\|f^\star\|_K^2 = \sum_{i=1}^{u+\ell} \sum_{j=1}^{u+\ell} w_i^\top K(\mathbf{x}_i, \mathbf{x}_j) w_j. \tag{4.17}$$

To enforce the smoothness of the representation across the different views, the regularizer

$$\|f\|_M^2 = \sum_{\alpha=1}^{m} \sum_{i,j=1}^{u+\ell} f^\alpha(x_i^\alpha) M_{ij}^\alpha f^\alpha(x_j^\alpha). \tag{4.18}$$

is introduced. Through the weights M_{ij}^α, we can easily handle situations of either agreement or discrepancy within the α-th view $f^\alpha(x_i^\alpha)$ and $f^\alpha(x_j^\alpha)$ for different data points. Mathematically, this is done by fixing M^α to be the graph Laplacian operator [Bel+06; Min+13].

Comments on the usage of the Huber Loss. As we argued, the Huber loss is a trade-off between L^1 and L^2 penalty functions. Therefore, it seems natural to apply such function to the case of scalar regression tasks for which the two aforementioned losses are broadly applied in the literature. In addition,

the Huber loss is a classical estimator used for robust estimation problems. By design, it avoids bigger errors to be over-penalized by means of a linear growth in the error rates which allows the model to better recover from them as opposed to a fully quadratic cost function. Keep in mind that the switch from the quadratic to the linear growth of the Huber loss is controlled by means of the threshold value ξ. In this Chapter, we propose to adaptively learn ξ from the data so that it can represents a sort of intermediate level of noise present in our annotations. Those annotations which exceed such threshold are automatically characterized as outliers and therefore removed from the learning pipeline and, in such a case, we perform unsupervised learning of the corresponding data which are used as unannotated instances to support the learning of the geometrical inner structure of the data. All the remaining instances, which show a level of corruption which is less or equal to this threshold value are recognized as clean data from which a model can be proficiently learned. Thanks to our implementation, such two-fold nature of the Huber loss seems a very straightforward and natural tool to handle label noise.

The optimization framework (4.4) is very general and applicable to a broad class of particular cases, including single-viewed ($m = 1$), fully supervised ($u = 0$) and classical regularization frameworks ($\gamma = 0$). In such general framework, the following result presents our novel solution to perform an exact optimization of the Huber loss (4.2).

Theorem 6 (General solution for Huber loss multi-view manifold regularization regression). *For any $\xi > 0$, the coefficients $\mathbf{w} = [w_1, \ldots, w_{u+\ell}]^\top$ defining the solution (4.16) of problem (4.15) are given by*

$$2\ell\lambda w_i + 2\ell\gamma \sum_{j,h=1}^{u+\ell} M_{ij} K(\mathbf{x}_j, \mathbf{x}_h) w_h = \\ = \begin{cases} -\xi c & \text{if } i \in L_+[\mathbf{w}, \xi] \\ \left(y_i - \sum_{j=1}^{u+\ell} c^\top K(\mathbf{x}_i, \mathbf{x}_j) w_j \right) c & \text{if } i \in L_0[\mathbf{w}, \xi] \\ +\xi c & \text{if } i \in L_-[\mathbf{w}, \xi] \\ 0 & \text{otherwise} \end{cases} \tag{4.19}$$

where $\lambda, \gamma > 0$, we set $M_{ij} = \operatorname{diag}(M_{ij}^1, \ldots, M_{ij}^m)$, and

$$L_+[\mathbf{w}, \xi] = \left\{ i \leq \ell \colon \sum_{j=1}^{u+\ell} c^\top K(\mathbf{x}_i, \mathbf{x}_j) w_j \geq y_i + \xi \right\}, \tag{4.20}$$

$$L_0[\mathbf{w}, \xi] = \left\{ i \leq \ell \colon \left| \sum_{j=1}^{u+\ell} c^\top K(\mathbf{x}_i, \mathbf{x}_j) w_j - y_i \right| < \xi \right\}, \tag{4.21}$$

$$L_-[\mathbf{w}, \xi] = \left\{ i \leq \ell \colon \sum_{j=1}^{u+\ell} c^\top K(\mathbf{x}_i, \mathbf{x}_j) w_j \leq y_i - \xi \right\}. \tag{4.22}$$

Proof. while applying the Representer Theorem [SS02] to cast the optimization problem into a minimizing on the coefficients $\mathbf{w}$ defining $f^\star$ in (4.16). Precisely,

implementing it in (4.4) yields

$$\mathcal{J}_{\lambda,\gamma}(\mathbf{w}) = \frac{1}{\ell}\sum_{i=1}^{\ell} H_\xi\left(y_i - \sum_{j=1}^{u+\ell} c^\top K(\mathbf{x}_i,\mathbf{x}_j)w_j\right) +$$
$$+\lambda\sum_{j,k=1}^{u+\ell} w_j^\top K(\mathbf{x}_j,\mathbf{x}_k)w_k + \gamma\sum_{i,j=1}^{u+\ell}\sum_{h,k=1}^{u+\ell} w_h^\top K(\mathbf{x}_h,\mathbf{x}_i)M_{ij}K(\mathbf{x}_j,\mathbf{x}_k)w_j. \tag{4.23}$$

Theoretically, minimizing $J_{\lambda,\gamma}$ over the RKHS $\mathcal{S}_K$ is fully equivalent to minimizing $\mathcal{J}_{\lambda,\gamma}$ with respect to $\mathbf{w}$, being the latter approach computationally convenient because, for this purpose, the optimization domain $\mathbb{R}^{m(u+\ell)}$ is preferable to an infinite-dimensional functional space. Notice that each addend of $\mathcal{J}_{\lambda,\gamma}$ is differentiable with respect to $\mathbf{w}$. Indeed, by computing the derivate of (4.2)

$$H'_\xi(y) = \begin{cases} -\xi & \text{if } y \le -\xi \\ y & \text{if } |y| \le \xi \\ +\xi & \text{if } y \ge \xi, \end{cases} \tag{4.24}$$

and $\|f\|_K^2$ and $\|f\|_M^2$ in (4.23) are replaced by differentiable polynomials in $w_1,\dots,w_{u+\ell}$. Thus, for any $p = 1,\dots,u+\ell$ and $\eta = 1,\dots,m$, we compute $\frac{\partial \mathcal{J}_{\lambda,\gamma}}{\partial w_p^\eta}$, differentiating with respect to w_p^η, the η-th view of w_p. Then,

$$\frac{\partial \mathcal{J}_{\lambda,\gamma}}{\partial w_p^\eta} = -\frac{1}{\ell}\sum_{i=1}^{\ell} H'_\xi\left(y_i - \sum_{j=1}^{u+\ell} c^\top K(\mathbf{x}_i,\mathbf{x}_j)w_j\right)\sum_{h=1}^{u+\ell}\sum_{\alpha=1}^{m} c^\alpha \kappa^\alpha(x_i^\alpha,x_h^\alpha)\frac{\partial w_h^\alpha}{\partial w_p^\eta} +$$
$$\lambda\sum_{j,k=1}^{u+\ell}\sum_{\alpha=1}^{m}\left[\frac{\partial w_j^\alpha}{\partial w_p^\eta}\kappa^\alpha(x_j^\alpha,x_k^\alpha)w_k^\alpha + w_j^\alpha\kappa^\alpha(x_j^\alpha,x_k^\alpha)\frac{\partial w_k^\alpha}{\partial w_p^\eta}\right] +$$
$$\gamma\sum_{i,j=1}^{u+\ell}\sum_{h,k=1}^{u+\ell}\sum_{\alpha=1}^{m}\frac{\partial w_h^\alpha}{\partial w_p^\eta}\kappa^\alpha(x_i^\alpha,x_h^\alpha)M_{ij}^\alpha\kappa^\alpha(x_j^\alpha,x_k^\alpha)w_k^\alpha +$$
$$\gamma\sum_{i,j=1}^{u+\ell}\sum_{h,k=1}^{u+\ell}\sum_{\alpha=1}^{m}\kappa^\alpha(x_i^\alpha,x_h^\alpha)w_h^\alpha M_{ij}^\alpha\kappa^\alpha(x_j^\alpha,x_k^\alpha)\frac{\partial w_k^\alpha}{\partial w_p^\eta}. \tag{4.25}$$

In order to simplify (4.25), we can apply the relationship $\frac{\partial w_i^\alpha}{\partial w_j^\eta} = \delta_{\alpha\eta}\delta_{ij}$, valid for any $\alpha,\beta = 1,\dots,m$ and $i,j = 1,\dots,u+\ell$, where, for any integers m,n, δ_{mn} is the Kronecker delta and $\delta_{mn} = 1$ if $m = n$, while, otherwise, $\delta_{mn} = 0$ if

$m \neq n$. Thus,

$$\frac{\partial \mathcal{J}_{\lambda,\gamma}}{\partial w_p^\eta} = -\frac{1}{\ell}\sum_{i=1}^{\ell} H'_\xi\left(y_i - \sum_{j=1}^{u+\ell} c^\top K(\mathbf{x}_i,\mathbf{x}_j)w_j\right)\sum_{h=1}^{u+\ell}\sum_{\alpha=1}^{m} c^\alpha \kappa^\alpha(x_i^\alpha, x_h^\alpha)\delta_{\alpha\eta}\delta_{hp} + \lambda\sum_{j,k=1}^{u+\ell}\sum_{\alpha=1}^{m}\left[\delta_{\alpha\eta}\delta_{jp}\kappa^\alpha(x_j^\alpha, x_k^\alpha)w_k^\alpha + w_j^\alpha\kappa^\alpha(x_j^\alpha, x_k^\alpha)\delta_{\alpha\eta}\delta_{kp}\right] + \gamma\sum_{i,j=1}^{u+\ell}\sum_{h,k=1}^{u+\ell}\sum_{\alpha=1}^{m}\delta_{\alpha\eta}\delta_{hp}\kappa^\alpha(x_i^\alpha, x_h^\alpha)M_{ij}^\alpha\kappa^\alpha(x_j^\alpha, x_k^\alpha)w_k^\alpha + \gamma\sum_{i,j=1}^{u+\ell}\sum_{h,k=1}^{u+\ell}\sum_{\alpha=1}^{m}\kappa^\alpha(x_i^\alpha, x_h^\alpha)w_h^\alpha M_{ij}^\alpha\kappa^\alpha(x_j^\alpha, x_k^\alpha)\delta_{\alpha\eta}\delta_{kp}. \tag{4.26}$$

By exploiting the properties of Kronecker delta, inside a summation over the index i, δ_{ij} discards all the addends except to j. Then, we can rewrite equation (4.26) obtaining

$$\frac{\partial \mathcal{J}_{\lambda,\gamma}}{\partial w_p^\eta} = -\frac{1}{\ell}\sum_{i=1}^{\ell} H'_\xi\left(y_i - \sum_{j=1}^{u+\ell} c^\top K(\mathbf{x}_i,\mathbf{x}_j)w_j\right) c^\eta\kappa^\eta(x_i^\eta, x_p^\eta) + \lambda\sum_{k=1}^{u+\ell}\kappa^\eta(x_p^\eta, x_k^\eta)w_k^\eta + \lambda\sum_{j=1}^{u+\ell} w_j^\eta\kappa^\eta(x_j^\eta, x_p^\eta) + \gamma\sum_{i,j=1}^{u+\ell}\sum_{k=1}^{u+\ell}\kappa^\eta(x_i^\eta, x_p^\eta)M_{ij}^\eta\kappa^\eta(x_j^\eta, x_k^\eta)w_k^\eta + \gamma\sum_{i,j=1}^{u+\ell}\sum_{h=1}^{u+\ell}\kappa^\eta(x_i^\eta, x_h^\eta)w_h^\eta M_{ij}^\eta\kappa^\eta(x_j^\eta, x_p^\eta). \tag{4.27}$$

To rearrange equation (4.27), we can exploit the functional symmetry of both Mercer kernels $\kappa^1, \ldots, \kappa^m$ and linear operators $M^1, \ldots, M^m$. Then, one sees

$$\frac{\partial \mathcal{J}_{\lambda,\gamma}}{\partial w_p^\eta} = -\frac{1}{\ell}\sum_{i=1}^{\ell} H'_\xi\left(y_i - \sum_{j=1}^{u+\ell} c^\top K(\mathbf{x}_i,\mathbf{x}_j)w_j\right) c^\eta\kappa^\eta(x_i^\alpha, x_p^\alpha) + 2\lambda\sum_{k=1}^{u+\ell} w_k^\eta\kappa^\eta(x_p^\alpha, x_k^\alpha) + 2\gamma\sum_{i,j=1}^{u+\ell}\sum_{k=1}^{u+\ell}\kappa^\eta(x_i^\alpha, x_p^\alpha)M_{ij}^\eta\kappa^\eta(x_j^\alpha, x_k^\alpha)w_k^\eta. \tag{4.28}$$

After vectorizing with respect to $\eta = 1, \ldots, m$, the derivative $\frac{\partial \mathcal{J}_{\lambda,\gamma}}{\partial w_p}$ equals to

$$-\frac{1}{\ell}\sum_{i=1}^{\ell} H'_{\xi}\left(y_i - \sum_{j=1}^{u+\ell} c^{\top} K(\mathbf{x}_i, \mathbf{x}_j) w_j\right) K(\mathbf{x}_p, \mathbf{x}_i) c +$$
$$+2\lambda \sum_{k=1}^{u+\ell} K(\mathbf{x}_p, \mathbf{x}_k)\omega_k + 2\gamma \sum_{i,j=1}^{u+\ell}\sum_{k=1}^{u+\ell} K(\mathbf{x}_p, \mathbf{x}_i) M_{ij} K(\mathbf{x}_j, \mathbf{x}_k) w_k. \tag{4.29}$$

Expression (4.29) rewrites $\sum_{i=1}^{u+\ell} K(\mathbf{x}_p, \mathbf{x}_i)\psi_i$, once we define ψ_i, for any i,

$$\psi_i = -\mathbf{I}(i \le \ell)\frac{1}{\ell} H'_{\xi}\left(y_i - \sum_{j=1}^{u+\ell} c^{\top} K(\mathbf{x}_i, \mathbf{x}_j) w_j\right) c +$$
$$2\lambda\omega_i + 2\gamma \sum_{j,h=1}^{u+\ell} M_{ij} K(\mathbf{x}_j, \mathbf{x}_h) w_h, \tag{4.30}$$

where the indicator function $\mathbf{I}$ is conditionally defined to be $\mathbf{I}(i \le \ell) = 1$ if $i \le \ell$ and $\mathbf{I}(i \le \ell) = 0$ if $i > \ell$.

If we set $\psi_1 = \cdots = \psi_{u+\ell} = 0$, then, from equation (4.28), $\frac{\partial \mathcal{J}_{\lambda,\gamma}}{\partial w_1} = \cdots = \frac{\partial \mathcal{J}_{\lambda,\gamma}}{\partial w_{u+\ell}} = 0$ and this leads to a solution of (4.4). But, this is the only solution we have since, as motivated, the optimization problem (4.4) has unique solution thanks to Representer Theorem [SS02]. Then, the previous discussion ensures that, globally, the two systems of equations are totally equivalent since, for every $i = 1, \ldots, u + \ell$,

$$\psi_i = 0 \quad \text{if and only if} \quad \frac{\partial \mathcal{J}_{\lambda,\gamma}}{\partial w_i} = 0. \tag{4.31}$$

Hence, the optimization of (4.4) can be done by solving

$$2\ell\lambda w_i + 2\ell\gamma \sum_{j,h=1}^{u+\ell} M_{ij} K(\mathbf{x}_j, \mathbf{x}_h) w_h =$$
$$H'_{\xi}\left(y_i - \sum_{j=1}^{u+\ell} c^{\top} K(\mathbf{x}_i, \mathbf{x}_j) w_j\right) c \tag{4.32}$$

for $i = 1, \ldots, \ell$; and, when $i = \ell + 1, \ldots, u + \ell$,

$$2\lambda w_i + 2\gamma \sum_{j,h=1}^{u+\ell} M_{ij} K(\mathbf{x}_j, \mathbf{x}_h) w_h = 0. \tag{4.33}$$

Substitute equation (4.24) into (4.12). Then, for $i = 1, \dots, \ell$,

$$2\ell\lambda w_i + 2\ell\gamma \sum_{j,h=1}^{u+\ell} M_{ij} K(\mathbf{x}_j, \mathbf{x}_h) w_h = \begin{cases} -\xi c & \text{if } y_i - \sum_{j=1}^{u+\ell} c^\top K(\mathbf{x}_i, \mathbf{x}_j) w_j \leq -\xi \\ \left(y_i - \sum_{j=1}^{u+\ell} c^\top K(\mathbf{x}_i, \mathbf{x}_j) w_j \right) c & \text{if } \left| y_i - \sum_{j=1}^{u+\ell} c^\top K(\mathbf{x}_i, \mathbf{x}_j) w_j \right| \leq \xi \\ +\xi c & \text{if } y_i - \sum_{j=1}^{u+\ell} c^\top K(\mathbf{x}_i, \mathbf{x}_j) w_j \geq \xi. \end{cases} \tag{4.34}$$

If one defines the following set of indexes

$$L_+ = L_+[\mathbf{D}, \mathbf{w}, \xi] = \left\{ i \leq \ell : \sum_{j=1}^{u+\ell} c^\top K(\mathbf{x}_i, \mathbf{x}_j) w_j \geq y_i + \xi \right\},$$

$$L_0 = L_0[\mathbf{D}, \mathbf{w}, \xi] = \left\{ i \leq \ell : \left| \sum_{j=1}^{u+\ell} c^\top K(\mathbf{x}_i, \mathbf{x}_j) w_j - y_i \right| < \xi \right\},$$

$$L_- = L_-[\mathbf{D}, \mathbf{w}, \xi] = \left\{ i \leq \ell : \sum_{j=1}^{u+\ell} c^\top K(\mathbf{x}_i, \mathbf{x}_j) w_j \leq y_i - \xi \right\},$$

equation (4.34) therefore becomes

$$2\ell\lambda w_i + 2\ell\gamma \sum_{j,h=1}^{u+\ell} M_{ij} K(\mathbf{x}_i, \mathbf{x}_h) w_h = \begin{cases} -\xi c & \text{if } i \in L_+ \\ \left(y_i - \sum_{j=1}^{u+\ell} c^\top K(\mathbf{x}_i, \mathbf{x}_j) w_j \right) c & \text{if } i \in L_0 \\ +\xi c & \text{if } i \in L_- \end{cases} \tag{4.35}$$

The thesis follows as the straightforward combination of equations (4.35) and (4.13). □

The sets (4.20) and (4.22) collect those labeled data points $\mathbf{x}_i$ for which the prediction $\sum_{j=1}^{u+\ell} c^\top K(\mathbf{x}_i, \mathbf{x}_j) w_j$ over- or under-estimate the actual value y_i, respectively. In (4.21), the absolute error between the prediction and y_i is lower than ξ.

Once equation (4.19) is solved, we obtain the coefficients w_j which allow to compute $f^\star$ in (4.16), obtaining the exact minimizer of the problem (4.4). Actually, solving (4.19) can be easily done - it's just a linear system - *if* the sets (4.20), (4.21) and (4.22) are known. However, in general, this is not the case, since they actually depend on the solution $\mathbf{w}$ we are looking for.

In order to solve this problem and allow to use (4.19) to exactly optimize (4.4), we propose the Huber Loss Regression (HLR) algorithm in Section 4.2.1.

4.2.1 The Huber Loss Regression (HLR) algorithm

From the definition of the sets L_0, L_+ and L_-, it is easy to see that, in the case $\xi \to \infty$, (4.20) and (4.22) are empty, while (4.21) is equal to $\{1, \dots, \ell\}$. Therefore, in the case of an infinite ξ value, we are actually able to exactly solve (4.19), which reduces to the linear system

$$2\ell\lambda w_i + 2\ell\gamma \sum_{j,h=1}^{u+\ell} M_{ij} K(\mathbf{x}_j, \mathbf{x}_h) w_h = \left(y_i - \sum_{j=1}^{u+\ell} c^\top K(\mathbf{x}_i, \mathbf{x}_j) w_j \right) c, \tag{4.36}$$

$i = 1, \dots, u + \ell$, which corresponds to replace H_ξ in (4.4) with a quadratic loss. Once the solution $\mathbf{w}^{(0)}$ of (4.36) is computed,

$$\xi^{(0)} = \max_{i=1,\dots,\ell} \left| \sum_{j=1}^{u+\ell} c^\top K(\mathbf{x}_i, \mathbf{x}_j) w_j^{(0)} - y_i \right| \tag{4.37}$$

represents the maximum absolute error within the labeled training set. Then, since $L_0[\mathbf{w}^{(0)}, \xi^{(0)}] = \{1, \dots, \ell\}$, $\mathbf{w}^{(0)}$ solves (4.19) with $\xi = \xi^{(0)}$, that is, we can exactly optimize the Huber loss $H_{\xi^{(0)}}$ in (4.4).

Since $\xi^{(0)}$ quantifies the error of our model, as to improve the learning stage of $f^\star$, it seems natural to (try to) learn a better solution $\mathbf{w}^{(1)}$ in the sense of a lower $\xi^{(1)} < \xi^{(0)}$ value. This is done by, first, computing $\widetilde{\xi}^{(1)}$ by manually reducing $\xi^{(0)}$ of the fixed rate $\Delta\xi$. Second, we solve (4.19) while keeping unchanged the sets L_+, L_0 and L_-: this step produces a coefficient vector $\mathbf{w}^{(1)}$. Third, we can update the maximum of the absolute error scored by our model as well as (4.20), (4.21) and (4.22). As before, we can prove that $\mathbf{w}^{(1)}$ is the actual solution of (4.19) for $\xi = \xi^{(1)}$.

The aforementioned idea is the core of our Huber Loss Regression (HLR), presented in Algorithm 5. Precisely, HLR is a refinement scheme, where, for any $\tau = 0, \dots, T$, we are able to jointly learn from the data a novel threshold value

$$\xi^{(\tau)} = \max_{i \in L_0} \left| \sum_{j=1}^{u+\ell} c^\top K(\mathbf{x}_i, \mathbf{x}_j) w_j^{(\tau)} - y_i \right|, \tag{4.38}$$

and compute $\mathbf{w}^{(\tau)}$ by optimizing (4.19) with $\xi = \xi^{(\tau)}$. The latter statement is showed through the following result.

Proposition 2. *For any $\tau = 0, 1, \dots, T$, the coefficients $\mathbf{w}^{(\tau)}$ satisfy (4.19) with $\xi = \xi^{(\tau)}$, where*

$$\xi^{(\tau)} = \max_{i \le \ell} \left| \sum_{j=1}^{u+\ell} c^\top K(\mathbf{x}_i, \mathbf{x}_j) w_j^{(\tau)} - y_i \right|. \tag{4.39}$$

Proof. It is enough to show that, for any $\tau = 0, 1, \dots, T$, we get $L_0[\mathbf{D}^{(\tau)}, \mathbf{w}^{(\tau)}, \xi^{(\tau)}] = \{1, \dots, \ell\}$. Let's go by induction.

For $\tau = 0$, as mentioned before, $\mathbf{w}^{(0)}$ solves (4.19) for $\xi = +\infty$ since, for any dataset $\mathbf{D}$ and any vector $\mathbf{w}$ of coefficients $w_1, \dots, w_{u+\ell}$, we have $L_0[\mathbf{D}, \mathbf{w}, +\infty] = \{1, \dots, \ell\}$. Since $\xi^{(0)}$ represents the maximum absolute error inside the training

set $\mathbf{D}^{(0)} = \mathbf{D}$ when the solution of the optimization problem is specified by $\mathbf{w}^{(0)}$, we have $L_0[\mathbf{z}, \mathbf{w}^{(0)}, \xi^{(0)}] = \{1, \dots, \ell\}$. So the thesis is proved for $\tau = 0$.

Now, let's assume that (4.39) holds for the $(\tau-1)$-th refinement and we prove it for the τ-th one. As a consequence,

$$L_0[\mathbf{D}^{(\tau-1)}, \mathbf{w}^{(\tau-1)}, \xi^{(\tau-1)}] = \{1, \dots, \ell\} \tag{4.40}$$

and we must show that the same relation is valid also for τ. Once computed $\widetilde{\mathbf{w}}^{(\tau)}$, we do not discard y_i from the training data $\mathbf{D}^{(\tau-1)}$ if and only if $\left|\sum_{j=1}^{u+\ell} c^\top K(\mathbf{x}_i, \mathbf{x}_j)\widetilde{w}_j^{(\tau)} - y_i\right| \leq \widetilde{\xi}^{(\tau)}$. Since the algorithm requires to compute $\mathbf{w}^{(\tau)}$ by permuting $\mathbf{w}^{(\tau-1)}$ in a way that the elements $w_j^{(\tau-1)}$ with $j \in L_0[\mathbf{D}^{(\tau-1)}, \widetilde{\mathbf{w}}^{(\tau)}, \widetilde{\xi}^{(\tau)}]$ occupy the first entries, we have $\left|\sum_{j=1}^{u+\ell} c^\top K(\mathbf{x}_i, \mathbf{x}_j)w_j^{(\tau)} - y_i\right| \leq \widetilde{\xi}^{(\tau)}$ thanks to the assumption (4.40). Since $\xi^{(\tau)}$ is defined as the maximum of a finite set of elements all bounded by $\widetilde{\xi}^{(\tau)}$, we conclude

$$\xi^{(\tau)} = \max_{i=1,\dots,\ell} \left|\sum_{j=1}^{u+\ell} c^\top K(\mathbf{x}_i, \mathbf{x}_j)w_j^{(\tau)} - y_i\right| \leq \widetilde{\xi}^{(\tau)}. \tag{4.41}$$

From the previous relation and from the definition of the set L_0, we obtain $L_0[\mathbf{D}^{(\tau)}, \mathbf{w}^{(\tau)}, \xi^{(\tau)}] = \{1, \dots, \ell\}$. □

Thank to Proposition 2, for any $\tau = 0, \dots, T$, we can ensure that our proposed HLR scheme is able to perform exact optimization for the Huber loss, whose threshold ξ is set to $\xi^{(\tau)}$ automatically. Therefore, within the τ-th refinement, we are learning the updated value $\xi^{(\tau)}$ for ξ, being able to exactly optimize the regularized Huber loss $H_{\xi^{(\tau)}}$ in (4.4). This aspect displays our originality with respect to classical optimization schemes [MM00; AZ05; LLZ11; Kha+13] which only provide an approximated solution to the problem. Moreover, in addition to be data-driven, we obtain another favorable property of the selected value $\xi^{(\tau)}$ for the maximum absolute error paid inside the training set. Indeed, we can show that from the τ-th to the $\tau+1$-th refinement, such error strictly decreases - in formulæ $\xi^{(\tau)} > \xi^{(\tau+1)}$. The latter claim is proved by the following proposition.

Proposition 3. *The sequence $\xi^{(0)}, \xi^{(1)}, \dots, \xi^{(T)}$ is monotonically strictly decreasing.*

Proof. In formulæ, we want to show that $\xi^{(0)} > \xi^{(1)} > \cdots > \xi^{(T)}$. In order to prove monotonicity, we fix an arbitrary refinement $\tau = 1, \dots, T$ and our goal is to show $\xi^{(\tau)} < \xi^{(\tau-1)}$. Directly using (4.41), we have

$$\xi^{(\tau)} \leq \widetilde{\xi}^{(\tau)}. \tag{4.42}$$

By definition of $\widetilde{\xi}^{(\tau)}$,

$$\widetilde{\xi}^{(\tau)} = \xi^{(\tau-1)} - \Delta\xi, \tag{4.43}$$

and, since $\Delta\xi > 0$, then

$$\xi^{(\tau-1)} - \Delta\xi < \xi^{(\tau-1)}. \tag{4.44}$$

Algorithm 5: HLR algorithm pseudocode

Input: **D** dataset, $M^1, \ldots, M^m$ graph Laplacians, $\lambda, \gamma > 0$ regularizing parameters, $\Delta\xi > 0$ updating rate, maximum number T of refinements.
Output: Coefficients vector $\mathbf{w}^\star \in \mathbb{R}^{m(u+\ell)}$.

begin
  Solve (4.36) with respect to $\mathbf{w}^{(0)}$
  Compute $\xi^{(0)}$ as in (4.37), $L_0 = \{1, \ldots, \ell\}$, $L_+ = L_- = \varnothing$
  for $\tau = 1, \ldots, T$ **do**
    Compute $\widetilde{\xi}^{(\tau)} = \xi^{(\tau-1)} - \Delta\xi$.
    Using the precomputed sets L_0, L_+, L_-, solve (4.19) with respect to $\mathbf{w}^{(\tau)}$ with $\xi = \widetilde{\xi}^{(\tau)}$.
    if L_0 *is empty is empty* **then**
      return Return $\mathbf{w}^\star := \mathbf{w}^{(\tau-1)}$.
    else
      Compute $\xi^{(\tau)}$ using (4.38).
      Update (4.20), (4.21) and (4.22) for $\mathbf{w} = \mathbf{w}^{(\tau)}$ and $\xi = \xi^{(\tau)}$.
    end
  end
  return $\mathbf{w}^\star := \mathbf{w}^{(\tau)}$
end

Combining the equations (4.42), (4.43) and (4.44) we get

$$\xi^{(\tau)} < \xi^{(\tau-1)}. \tag{4.45}$$

The thesis follows after the generality of τ in (4.45). □

The latter theoretical result implements the idea of a sequential shrinking of the threshold ξ towards the convergence to an optimal value $\xi^{(T)}$ in the sense of the removal of those annotation which are either noisy or inconsistent for improving the learning of the regression map. Precisely, HLR is able to automatically select which output variables $y_1, \ldots, y_\ell$ are not advantageous to improve the learning stage of the regression function. Indeed, at each refinement, HLR scans the labeled training set $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_\ell, y_\ell)$, checking whether, for $i = 1, \ldots, \ell$,

$$\left| \sum_{j=1}^{u+\ell} c^\top K(\mathbf{x}_i, \mathbf{x}_j) w_j^{(\tau)} - y_i \right| \geq \xi^{(\tau)}. \tag{4.46}$$

Equation (4.46) means that, for $\mathbf{x}_i$, the prediction of HLR is suboptimal, since differing from the actual value y_i for more than $\widetilde{\xi}^{(\tau)}$. In such case, the algorithm automatically removes y_i from the dataset, assigning $\mathbf{x}_i$ to be unlabeled and trying to exploit it in terms of the geometrical information provided by comparing the value $f(\mathbf{x}_i)$ with the other $f(\mathbf{x}_j)$ by means of the view-specific graph Laplacians $M^1, \ldots, M^m$.

Overall, the computational cost of HLR is $O((T+1)m^2(u+\ell)^2)$, always remaining the same in the case of a either labeled or unlabeled usage of any training instance $\mathbf{x}_i$.

4.3 Robust multi-view regression: a statistical experimental baseline

This Section presents our empirical analysis of HLR. In Section 4.3.1, we compare our algorithm, with a state-of-the-art optimizer for convex objective functions. In Section 4.3.2, the HLR automatic unlabeling component is applied on noisy curve fitting and benchmarked against several approaches for learning with noisy labels. Section 4.3.3 compares HLR with popular regression methods on classical machine learning datasets. Finally, in Section 4.4, we consider the crowd counting application, validating our method against the state-of-the-art ones in the literature through several experiments on three benchmark datasets.

4.3.1 Comparison with the state-of-the-art convex solver

The proposed HLR leverages on an exact solution for optimizing the Huber loss, as opposed to the iterative solving of many paradigms [MM00; AZ05; LLZ11; Kha+13]. In order to experimentally check the potentialities of such aspect, we

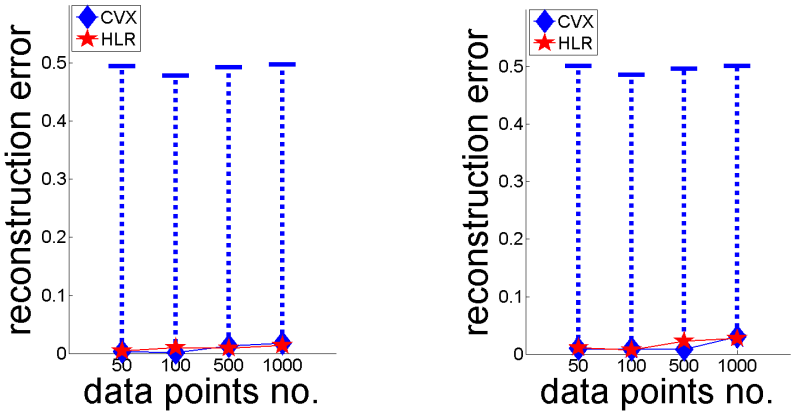


FIGURE 4.2: HLR versus CVX [GB08] in either a noise-free (left) or additive Gaussian noise setup (right). For CVX, after cross validating $\xi \in \{10^{-3}, 10^{-2}, \dots, 10^{3}\}$, the blue diamonds represent the best configuration resulting in the lowest reconstruction error, whose fluctuation is represented by means of the error bars.

compare HLR against CVX [GB08], the state-of-the-art optimization tool for convex problems. Precisely, by either exploiting HLR or CVX, we are able to optimize the same objective functional (4.4), consequently investigating which method is more efficient in terms of both reconstruction error and running time. Also, we are able to inspect HLR in the automatic pipeline of learning ξ against a standard cross-validation procedure which is necessary for CVX.

To do so, we consider the linear regression problem to predict $y \in \mathbb{R}$ from $\mathbf{x} \in \mathbb{R}^{10}$ where $y = \beta^{\top}\mathbf{x}$, $\beta = [1/10, \dots, 1/10]^{\top}$. We randomly generate $n = 50, 100, 500$ and 1000 samples $\mathbf{x}$ from a uniform distribution over the unit 10-dimensional hypercube $[0,1] \times \cdots \times [0,1]$. As a further experiment, we introduce some outliers to the model which becomes $y = \beta^{\top}\mathbf{x} + \epsilon$, where the additive noise ϵ is distributed according to a zero-mean Gaussian with 0.1 variance. For HLR, $T = 1$ and $\Delta\xi = 0.1$, $\lambda = 10^{-2}$ and $\gamma = 10^{-3}$ are fixed. The performance of CVX and HLR are measured via the reconstruction error between the ground truth values and the predictions. Also, we monitor the computational running time of both.

The analysis of Figure 4.2 yields to the following comments.

- Numerically, our general solution shows a comparable performance with respect to classical iterative schemes in terms of reconstruction error.
- For both algorithms, the noise ϵ does not remarkably influence the reconstruction error: this is due to the robustness provided by the Huber loss.
- When ξ varies in $\{10^{-3}, 10^{-2}, \dots, 10^{3}\}$, the reconstruction error of CVX greatly fluctuates and justifies the variability of the blue error bars in Figure 4.2

Moreover, Table 4.1 clarifies that HLR is much faster: the runtime[1] of HLR is about a few seconds even if n grows while, for CVX, it sharply raises in the cases $n = 500$ and $n = 1000$. Also, it is worth nothing that the computational

[1]For all experiments, we used MATLAB R2015b on a Intel(R) Xeon(R) CPU X5650 @2.67 GHz $\times 2$ cores and 12 GB RAM.

	$n = 50$	$n = 100$	$n = 500$	$n = 1000$
CVX	1.0 ± 0.2	1.9 ± 0.3	36.0 ± 10.0	179.7 ± 4.2
HLR	**0.1**	**0.1**	**3.4**	**18.1**

TABLE 4.1: HLR versus CVX - computational running time (measured in seconds).

Noise level	1%	10%	25%	50%	75%
s	1	1	0.89	0.67	0.39
error	0.21	0.23	0.27	0.75	1.25

TABLE 4.2: Noisy curve fitting. Sørensen-Dice index s and reconstruction error for variable noise levels.

running time for CVX is averaged over all the cross-validating repetitions for the different ξ values, leading to the mean and standard deviation values reported in Table 4.1.

Globally, HLR provides low reconstruction errors as CVX, being superior to it in terms of 1) faster computation and 2) automatic learning ξ.

4.3.2 Evaluation of the auto-unlabeling component

In this Section, we evaluate the robustness provided by the HLR auto-unlabeling component resulted from the usage of the Huber loss. For this purpose, we consider a noisy curve fitting experiment and we also faced the problem of binary classification in a corrupted data regime.

Noisy curve fitting. As in Section 4.3.1, starting from the same linear model $y = \beta^\top \mathbf{x}$, we severely corrupted a random percentage of target data points by inverting their sign. It is a quite sensible change since each entry of $\mathbf{x}$ is uniformly distributed in $[0,1]$, being thus non-negative. Consequently, our algorithm should be able to recognize the negative data as outliers and automatically remove them from the training set. Such evaluation is performed through Table 4.2 where, for several noise rates, we report the reconstruction error while measuring whether the labels removed by HLR actually refers to corrupted inputs. For the latter, we employ the Sørensen-Dice index s to measure the amount of corrupted data effectively removed by the HLR. In formulæ,

$$s = \frac{2|\mathcal{C} \cap \mathcal{R}|}{|\mathcal{C}| + |\mathcal{R}|} \tag{4.47}$$

where the sets $\mathcal{C}$ and $\mathcal{R}$ collects the corrupted and removed data, respectively: $s \in [0,1]$ and spans from the worst overlap case ($s = 0$ since $\mathcal{C} \cap \mathcal{R} = \varnothing$) to the perfect one ($s = 1$ if $\mathcal{C} = \mathcal{R}$).

In Table 4.2, despite the increasing noise level, the reconstruction error is quite stable and only degrades at the highest noise levels. Additionally, when the noise level has a minor impact (1% and 10%), we get $s = 1$: the removal process is perfect and exactly all the corrupted labels are effectively removed. When percentages of noise increases (25%, 50%), we still have good overlapping

Method	*House*	*Air*	*Hydro*	*Wine*
GPR	Affine mean, mixture covariance type (linear + squared exponential)			
RR	$\alpha = 0.3$	$\alpha = 0.01$	$\alpha = 0.5$	$\alpha = 1$
K-nn	$K = 3$	$K = 3$	$K = 3$	$K = 5$
NN	$n_h = 3$	$n_h = 5$	$n_h = 7$	$n_h = 6$
SVR	$\epsilon = 0.003$ $C = 10$	$\epsilon = 0.005$ $C = 10$	$\nu = 0.003$ $C = 1$	$\nu = 0.01$ $C = 10$
HLR	$\lambda = 0.001, \gamma = 0.0001, \Delta\xi = 0.01, T = 3$			

TABLE 4.3: In addition to the parameters $\lambda, \gamma, \Delta\xi$ and T of HLR, we report the parameters/settings of other methods benchmarked on the UCI Machine Learning Repository experiments: the mean and covariance functions used for GPR, the regularizing parameter α for RR, the value of K neighbors considered, the number of neurons n_h in the hidden layer for NN and the ϵ/ν choices for SVR as well as the cost function C used.

measures. The final drop at 75% is coherent with the huge amount of noise (only 1 target out of 4 is not corrupted).

Learning with noisy labels. We want to benchmark HLR in handling noisy annotations adopting the protocol of [Nat+13]. Therein, binary classification is performed in the presence of random noise so that some of the positive and negative labels have been randomly flipped with a given probability. Precisely, following [Nat+13], we denote with ρ_+ the probability that the label of a positive sample is flipped from $+1$ to -1. In a similar manner, ρ_- quantifies the negative instances whose label is wrongly assigned to be -1. In [Nat+13], such problem is stated under a theoretical perspective, formulating some bounds for the generalization error and the empirical risk, as to guarantee the feasibility of the learning task even in such an extreme situation. Although interesting per se, such arguments are out of the scope of our work, where, instead, we compared HLR with the two methods proposed by [Nat+13]: a surrogate logarithmic loss function ($\widetilde{\ell}_{\log}$) and a variant of support vector machine algorithm, where the cost parameter is adapted depending on the training labels (C-SVM). In [Nat+13], $\widetilde{\ell}_{\log}$ and C-SVM were shown to outperform other methods devised for the identical task: the max-margin perceptron algorithm (PAM) [KW07], Gaussian herding (NHERD) [CL10] and random projection classifier (RP) [SR09]. All the aforementioned methods are compared with HLR where, as usually done for binary decision boundaries, we exploit the sign of the learnt regression function to perform classification. To ensure a fair comparison, we reproduce the same experimental protocol (Gunnar Raetsch's training/testing splits and data preprocessing for *Breast Cancer*, *Diabetes*, *Thyroid*, *German*, *Heart*, *Image* benchmark datasets[2]) and we compute the testing accuracy with respect to the clean distribution of labels [Nat+13].

From the experimental results reported in Table 4.4, HLR scored a strong performance. Indeed, despite some modest classification results on Thyroid an Diabetes datasets, HLR is able to beat the considered competitors, obtaining the best classification accuracy in the remaining 4 out of 6 ones (*Breast Cancer*, *German*, *Heart* and *Image*). Interestingly, this happens in both low and high

[2] http://theoval.cmp.uea.ac.uk/matlab

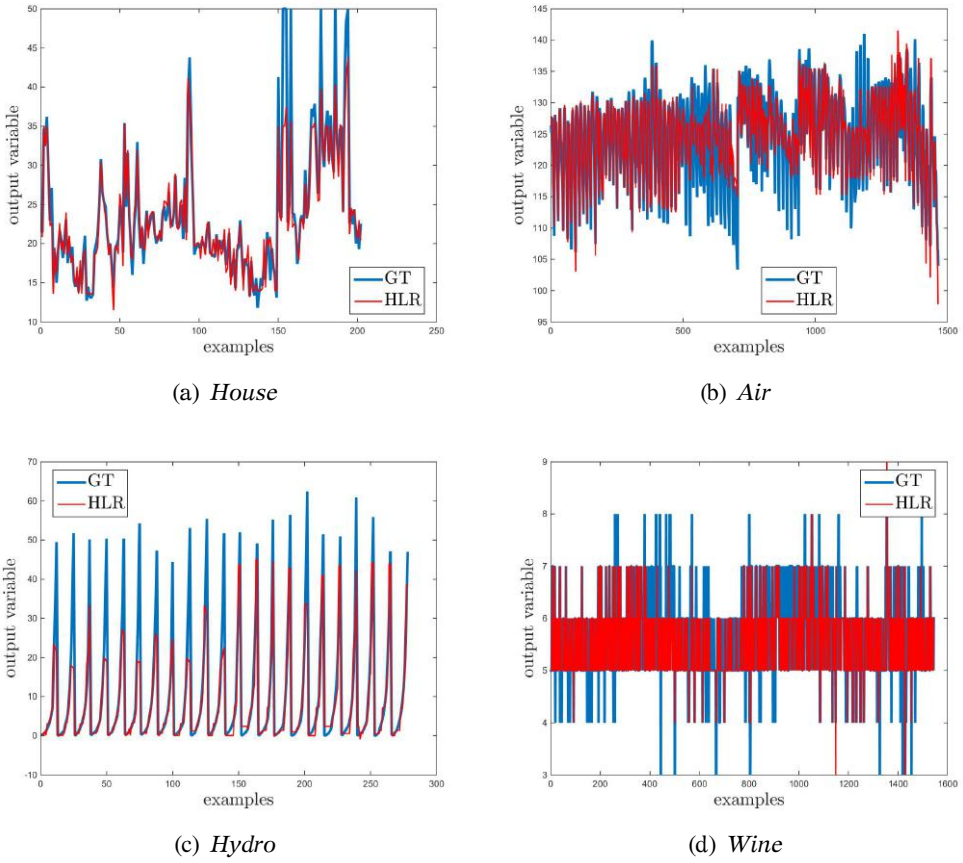
Dataset	ρ_+	ρ_-	$\widetilde{\ell}_{\log}$	C-SVM	PAM	NHERD	RP	HLR
Breast Cancer	0.2	0.2	70.12	67.85	69.34	64.90	69.38	**73.86**
	0.3	0.1	70.07	67.81	67.79	65.68	66.28	**71.90**
	0.4	0.4	**67.79**	**67.79**	67.05	56.50	54.19	59.12
Diabetes	0.2	0.2	**76.04**	66.41	69.53	73.18	75.00	75.39
	0.3	0.1	**75.52**	66.41	65.89	74.74	67.71	74.35
	0.4	0.4	65.89	65.89	65.36	**71.09**	62.76	66.37
Thyroid	0.2	0.2	87.80	94.31	**96.22**	78.49	84.02	92.43
	0.3	0.1	80.34	**92.46**	86.85	87.78	83.12	85.35
	0.4	0.4	83.10	66.32	70.98	**85.95**	57.96	84.15
German	0.2	0.2	71.80	68.40	63.80	67.80	62.80	**75.21**
	0.3	0.1	71.40	68.40	67.80	67.80	67.40	**72.86**
	0.4	0.4	67.19	**68.40**	67.80	54.80	59.79	62.54
Heart	0.2	0.2	**82.96**	61.48	69.63	**82.96**	72.84	81.53
	0.3	0.1	**84.44**	57.04	62.22	81.48	79.26	77.28
	0.4	0.4	57.04	54.81	53.33	52.59	68.15	**70.69**
Image	0.2	0.2	82.45	91.95	92.90	77.76	65.29	**92.92**
	0.3	0.1	82.55	89.26	89.55	79.39	70.66	**91.75**
	0.4	0.4	63.47	63.47	73.15	69.61	64.72	**82.38**

TABLE 4.4: Classification accuracies (in percentages) to compare different algorithms against HLR for the task of learning with noisy labels in a binary problem. We considered different rates ρ_+ and ρ_- to flip positive and negative examples, respectively. Best results in bold.

	House			*Air*		
Methods	MAE	MSE	MRE	MAE	MSE	MRE
GPR	**4.21(3)**	**41.00(3)**	**0.20(3)**	**4.47(2)**	**33.84(2)**	**0.03(1)**
RR	**3.79(1)**	**28.73(1)**	**16.03(2)**	**4.76(3)**	**37.61(3)**	**3.87(3)**
K-nn	5.91(6)	64.96(6)	22.64(6)	6.01(5)	65.57(6)	4.89(5)
NN	5.49(5)	56.97(5)	20.94(5)	6.56(6)	64.69(5)	5.32(6)
SVR	4.88(4)	51.55(4)	20.72(4)	4.93(4)	38.64(4)	3.99(4)
HLR	**4.13(2)**	**36.78(2)**	**0.15(1)**	**4.16(1)**	**30.20(1)**	**0.04(2)**

	Hydro		*Wine*		
Methods	MAE	MSE	MAE	MSE	MRE
GPR	**7.10(2)**	**118.3(3)**	**0.59(1)**	**0.72(1)**	**0.107(2)**
RR	**7.28(3)**	**113.9(2)**	**0.59(1)**	**0.72(1)**	**0.106(1)**
K-nn	9.08(6)	267.0(6)	0.61(5)	0.78(5)	0.108(4)
NN	8.32(5)	183.2(5)	0.86(6)	1.35(6)	0.161(6)
SVR	7.41(4)	143.8(4)	**0.59(1)**	**0.73(3)**	0.109(5)
HLR	**6.91(1)**	**110.8(1)**	0.61(4)	0.77(4)	**0.107(2)**

TABLE 4.5: Comparison of HLR against Gaussian Process Regression, Ridge Regression, K nearest neighbors, neural nets and support vector machine for regression. In bold, top three performing methods. In brackets, the relative ranking. For *Hydro*, since the target variable is sometimes (close to) zero, MRE metric diverges and therefore was not reported.



(a) *House* (b) *Air*

(c) *Hydro* (d) *Wine*

FIGURE 4.3: Ground truth (GT) compared with HLR prediction for UCI Machine Learning Repository datasets. Best viewed in colors.

noise levels: for instance, when $\rho_+ = \rho_- = 0.2$ on *Breast Cancer* and when $\rho_+ = \rho_- = 0.4$ on *Image*, respectively.

The results presented in this Section attest the auto-unlabeling HLR component to be able to effectively detect the presence of outliers data while, at the same time, guaranteeing an effective learning of the regression model.

4.3.3 Huber Loss Regression for Scalar Regression Problems

To compare the effectiveness of HLR in learning the regression map, in this Section, we benchmark on four datasets from the UCI Machine Learning Repository[3], we will focus on house pricing estimation (Boston Housing – *House*), physical simulations (AirFoil Self-Noise – *Air* and Yatch Hydrodynamics – *Hydro*) and agronomic quality control (*Wine*). We will briefly describe each of them.

House datasets predicts housing values in Boston suburbs. The dataset consists in 506 examples and 13 feature components which are either discrete (average number of rooms), binary (whether or not tracing bounds of Charles river) or continuous (pupil-teacher ratio by town). *Air* datasets address the problem of

[3] https://archive.ics.uci.edu/ml/datasets

physical simulations. It is provided by NASA and shows 1503 aerodynamic and acoustic acquisitions of two and three-dimensional air foil blade sections. A 6-dimensional feature vector encodes different size and thickness for blades, various wind tunnel speeds and angles of attack. The output variable is the sound pressure level measured in decibel. *Hydro* predicts the resistance of sailing yachts at the initial design stage, estimating the required propulsive power. Inputs provides hull dimensions an boat velocity (6 dimensional features, 308 instances). The output variable is the residuary resistance per unitary displacement. *Wine* dataset consists in 11-dimensional 1599 input instances (we only focused on red wine). The goal is predicting the ratings, given by a crew of sommeliers, as function of pH and alcohol/sulphates concentrations.

Over the aforementioned datasets, we compare Huber loss regression (HLR) against Gaussian process regression (GPR), ridge regression (RR), K nearest neighbors (K-nn), one-hidden-layer neural network (NN) and linear support vector machine for regression (SVR). For each method, the parameters setting are obtained after cross validation (see Table 4.3). For a fair comparison, we split each dataset in five equispaced folds and performing a leave-one-fold-out testing strategy. To give a comprehensive results on each datasets, we averaged the errors on each fold using one out of the following metrics: mean absolute error - MAE - mean squared error - MSE - and mean relative error - MRE.

Qualitative and quantitative analysis has been reported in Table 4.5 and Figure 4.3, respectively. Globally, HLR shows remarkable performances since outstanding the other methods in 5 cases out of 12. Those performances are also remarkable since they have been obtained with a fixed set of parameters, confirming the ductility of HLR (see Table 4.3). Indeed, despite GPR and RR scored comparable performance with respect to HLR, the regularizing parameter of RR has to be tuned and the parameters of the GPR has to be learnt in a maximum likelihood sense (mean function plus covariance kernel).

From this analysis, the low scored errors and the fixed parameter configuration make HLR outperforming many state-of-the-art approaches for scalar regression tasks.

4.4 Application to Crowd Counting

As a final test bed of our proposed framework, we address the crowd counting application, namely estimating the number of people in a real world environment using video data.

Crowd behaviour analysis has important actual applications both in security or event detection, and has been recently addressed by the computer vision community. In this context, *crowd counting* means estimating the number of people in a certain environment and profiling their dynamics over time. The lack of monitoring in crowding has potentially disastrous consequences. For example, one may remember Hillsborough and Heysel stadium tragedies (in 1985 and 1989, respectively), or the more recent (2010) love parade crowd crush in a music festival in Germany. Due to big amount of video surveillance data, human control of public gathering is unfeasible. On the other hand, automatic people counting is challenging due to low resolution videos,

inter-person occlusions, perspective distortion and more general visual ambiguities related, for example, to light variations [Lei+05],[KC09]. State-of-the-art methods adopt regression-based tecniques to learn a map between low level features and people count (*crowd density*), avoiding, as logical, explicit crowd detection or clustering [Cha+08],[Ma+04],[Mar+97],[Kon+06].

A consolidated taxonomy of crowd counting approaches identifies three main paradigms [Loy+13c]: *counting by detection*, *counting by clustering* and *counting by regression*.
In counting by detection, a classifier is trained to learn a model for a single person. This template is convolved with the original image and all the candidate positions for pedestrians are found. After a non maximum suppression, the number of detections will estimate crowd density [Lei+05]. As expected, this type of approaches is sensible to occlusions and deformable part models have been introduced to overcome this issue. For example, encoding shoulder region in a omega-shape pattern is effective in real-word applications [Li+08]. Counting by clustering is based on the extraction of coherent motion pattern from the crowd (*e.g.* with a KLT tracker [RB06]) and a successive clustering phase will give the number of people. Finally, counting by regression is a more straightforward apparatus. Indeed, one can directly estimate the number of people from image features without intermediate steps. Usually, the pipeline starts detecting in each frame a region of interest, while the effects of geometric distortion are removed with an homography [Ma+04]. Some features are extracted from the foreground and a regression map is trained. The works of [Dav+95] and [Ma+04] are based on crowd density modelling assuming a linear-affine relation between the number of people and the edge pixel number, once perspective distortion is corrected. While [Mar+97] extracted descriptors from mutual occurrences of grey levels, [Cha+08] fixed the most useful features in regression tasks which are mainly based on foreground area, pedestrian edges and texture statistics. Several methods exploited those features, like Bayesian regression models [CV12],[Cha+09a] or ridge regression [Che+12a],[Che+13a]. A group of recent papers [Tan+11a],[Loy+13a] tried to perform manifold learning to exploit geometric inner configuration of input data.

Differently from the aforementioned literature, we focus on Huber loss function which, to the best of our knowledge, has been never used for crowd counting. As in [Loy+13a], we devise an active learning component making HLR able to automatically control the number of unlabelled sample. Two are the main differences. First, it is an integrated component of our regression algorithm (and not a pre-processing step). Second, in training phase, we do not have to manually select how many examples has to be labelled, being the system able to automatically decide it.

4.4.1 Datasets used for Crowd Counting

Three benchmark datasets have been used to test the performances of our Huber loss regression. They are *MALL* [Che+12a], *UCSD* [Cha+08] and *PETS 2009* [Cha+09a]. For the sake of completeness, we will introduce each of them

FIGURE 4.4: An exemplar frame from *MALL* (left), *UCSD* (center) and *PETS 2009* (right) datasets, with relative region of interests (ROI) where crowd is assumed to wall through. For PETS 2009, three regions (R0, R1 and R2) are considered, and are represented in red, yellow and blue, respectively. Best viewed in colors.

MALL. More than 62.325 pedestrians have been recorded using a surveillance camera in a shopping centre. From the video, 2000 RGB images were extracted (resolution 320×240). In each image, crowd density varies from 13 to 53. The main challenges are related to shadows and reflections. Following the literature [Che+13a], our system is trained with the first 800 frames, and the remaining ones are left for testing.

UCSD. A hand-held camera recorded a campus outdoor scene composed by 2000 gray-scale frames of dimensions 238×158. The density grows from 11 to 46 and the total number of people is 49885. Environment changes are less severe, while geometric distorsion is sometimes a burden. Experiments are usually trained on frames $601\div1400$ [Che+12a].

PETS 2009. Within PETS 2009 workshop, a new dataset has been recorded from a British campus. Crowd counting experiments are carried out on sequences 13-57,13-59,14-03,14-06 from camera 1 [Cha+09a], and three regions of interests have been introduced (R0, R1 and R2 in Fig. 4.4). The overall amount of people is 21783: in each of the 768×576 RGB images, crowd density ranges between 0 and 42. Shadows and the appearance of both walking and running people are the main challenges.

4.4.2 Feature representations exploited

In our experiments, we consider the data as composed by $m = 3$ different views, each of them encoding a particular class of hand-crafted features which have been broadly applied within the panorama of crowd counting literature. These features can be categorized in three classes: *size*, *edge* and *texture*.

Size features refers to the magnitude of any interesting segments extracted from an image which are deemed to be relevant, such as the foreground pixel count.

This class of representation, depends on a continuous 2-dimensional mask $S = S(i,j)$, of the same size of the video frames, is learnt so that it can compensate from the perspective distortion. Precisely, the map quantifies how much correction should be applied to the pixel in position (i,j), in a manner

that, for instance, pixel that are far away from the camera should be weighted more than other ones which are more close to it. To a comprehensive presentation on how such mask in computed, please refer to [Rya+15]. Given S, one may compute the following statistics.

Size-1 **Area** - the total number of pixel which remains after subtracting the foreground (with pedestrians) by the background (without pedestrians).

Size-2 **Perimeter** - the number of neighboring pixels of the region which remains after subtracting the foreground (with pedestrians) by the background (without pedestrians).

Size-3 The **ratio** between Perimeter and Area

Size-4 **Oriented Perimeter** - the number of pixels which collaborate within the count of the **Perimeter** and are oriented by 0°, 30°, 60°, 90°, 120°, 150° with respect to the image center.

Size-5 **Blob count** - the number of connected components of the image which have more than k pixel in the segment (a typical value is $k = 10$)

Edge features refers to the relative change in pixel intensities across an image, and this is typically measured by means of a binary edge detector D - let's say computed with Canny algorithm [Can86]. Starting from D, the following statistics are extracted

Edge-1 The total number of **edge pixel** in D.

Edge-2 **Edge orientation** - in the form of 6-bin histogram which quantizes the interval $[0, \pi]$.

Edge-3 **Minkowski's number** - a scalar number which estimates the fractal dimension of the image and models the degree of "space-filling" of the edges (see [Mar+97] for more details).

Texture features refers to general descriptors of an image such as contrast and homogeneity. Such features are based on the grey-level co-occurrence matrix (GLCM) which is computed with the following steps. First, the image is quantized into g gray level (a typical value is $g = 16$) an the joint histogram of neighboring pixel values is estimated for the angles which varies between the values 0°, 45°, 90° and 135°. For each of those angular values, the following statistics are computed.

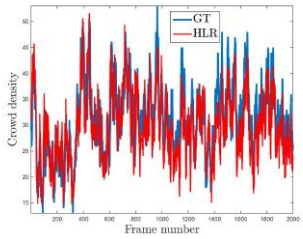
Texture-1 **Homogeneity** measures the smoothness of the texture

Texture-2 **Energy** computes the L^2 norm of the GLCM

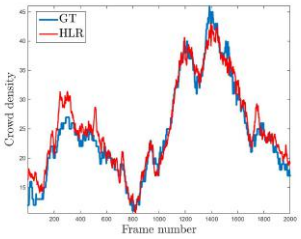
Texture-3 **Entropy** measures the randomness of the estimated probability distribution.

4.4.3 Crowd Counting Experiments: Qualitative Results of HLR

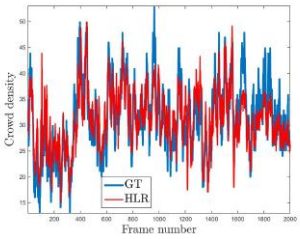
This Section illustrates the qualitative results depicted in Figures 4.5 and 4.6. Graphs 4.5(a) and 4.5(c) highlight HLR impressive performance on UCSD



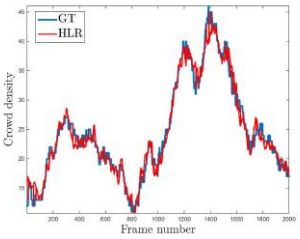
(a) MALL, first protocol



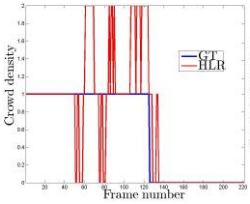
(b) USCD, first protocol



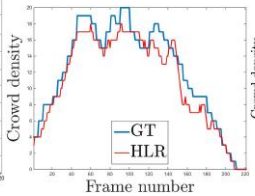
(c) MALL, second protocol



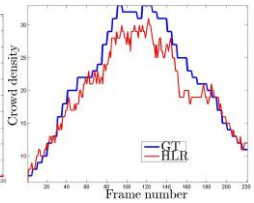
(d) UCSD, second protocol



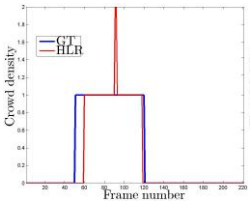
(e) 13 – 57 R0 left moving



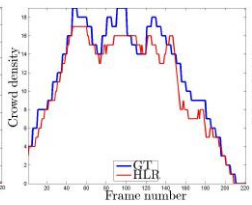
(f) 13 – 57 R0 right moving



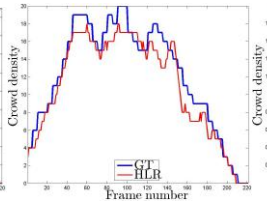
(g) 13 – 57 R0 total



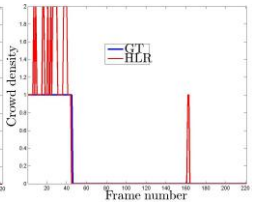
(h) 13 – 57 R1 left moving



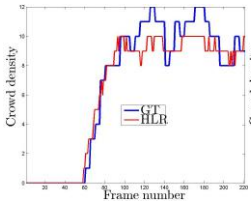
(i) 13 – 57 R1 right moving



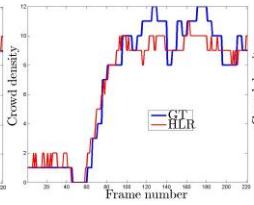
(j) 13 – 57 R1 total



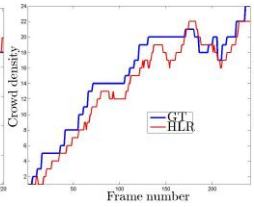
(k) 13 – 57 R2 left moving



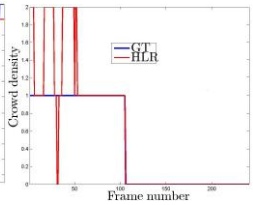
(l) 13 – 57 R2 right moving



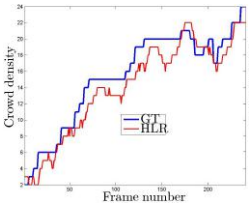
(m) 13 – 57 R2 total



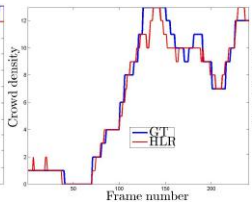
(n) 13 – 59 R0 left moving



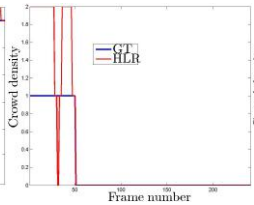
(o) 13 – 59 R0 right moving



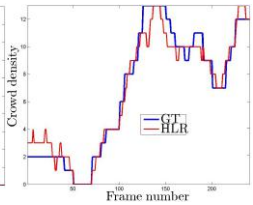
(p) 13 – 59 R0 total



(q) 13 – 59 R1 left moving

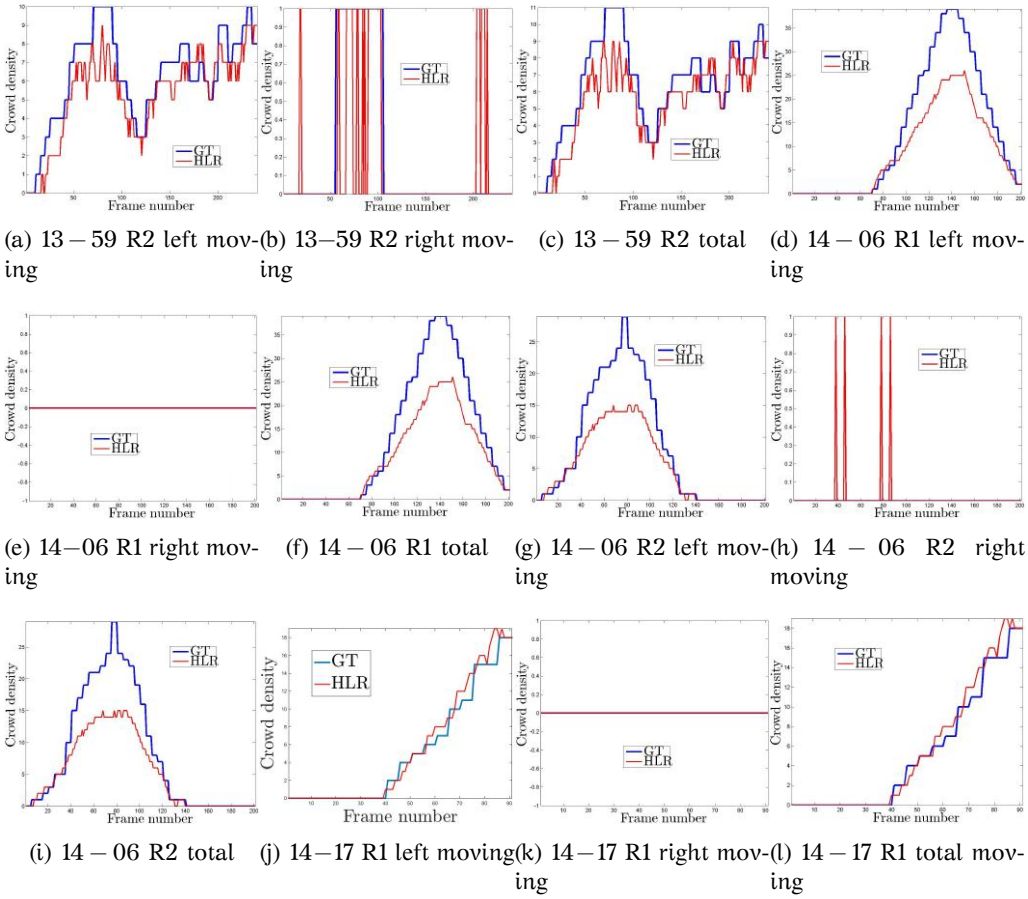


(r) 13 – 59 R1 right moving



(s) 13 – 59 R1 total

FIGURE 4.5: Qualitative results for HLR on crowd counting task. Ground truth crowd density (blue) is compared with HLR prediction (red). Graphs 4.5(a) and 4.5(c) refer to UCSD dataset, 4.5(b) and 4.5(d) to MALL, all the others sequences are drawn from PETS 2009 dataset. Best viewed in color.



(a) 13 – 59 R2 left moving (b) 13–59 R2 right moving (c) 13 – 59 R2 total (d) 14 – 06 R1 left moving

(e) 14–06 R1 right moving (f) 14 – 06 R1 total (g) 14 – 06 R2 left moving (h) 14 – 06 R2 right moving

(i) 14 – 06 R2 total (j) 14–17 R1 left moving (k) 14–17 R1 right moving (l) 14 – 17 R1 total moving

FIGURE 4.6: Qualitative results for HLR on the remaining sequences from PETS 2009, comparing ground truth crowd density (blue) with HLR prediction (red). Best viewed in color.

datasets on adopted protocols. The reconstructed profile of crowd density is extremely overlapping with the original one and very negligible local differences are appreciable. Switching to MALL dataset, plots 4.5(b) and 4.5(d) show more significant differences between ground truth (blue) and reconstructed profile (red), look in particular at the left part of figure 4.5(b).

All the other plots refer to the all the sequences from PETS 2009, each of them analysing either the whole crowd density or only people left/right moving. The main part of the graphs shows a remarkably good approximation of people number; see Figures 4.5(f), 4.5(m), 4.5(s), 4.6(a) and 4.6(l), for examples. Few plots, like those in Figures 4.6(d) and 4.6(g), present some peaks which HLR is not able to model in their actual growth. There are some particular cases in which only one person crosses the region of interest for a limited time (Figures 4.5(h), 4.5(k), 4.5(o), 4.5(r)). In these cases, crowd estimation has at maximum $+1$ or -1 difference with respect to the exact value and the big oscillations visible in the figure are only due the compressed scale on the ordered axis used for the view. Finally, there are two sequences in which crowd density is exactly 0, since no people was moving in that direction at that acquisition time, and HLR well recovers this situation, predicting the complete absence of crowd (Figures 4.6(e) and 4.6(k)).

4.4.4 Crowd Counting Experiments: Quantitative Results of HLR

In this Section, we illustrate the quantitative experimental results we obtain by benchmarking the proposed Huber Loss Regression (HLR) over UCSD, MALL and PETS 2009 benchmark dataset. Let us notice that, for the sake of a fair validation, we do not include in our baseline the results of deep learning methods [Wan+15a; Zha+15b; ORLS16; Zha+16b; Boo+16; Ste+16; Sam+17; Han+17; Xio+17] since those methods are usually trained on other bigger datasets and just transferred on the ones with are investigating right now. Moreover, applying feature learning in combination with multi-view learning seems not totally convincing due to the fact that, if one is allowed to learn the representation from the data itself, it is totally reasonable to assume that one can also learn how to circumvent the biases of the specific dataset, ultimately devising a feature representation which is much more powerful - since customized to the data - (and also more compact) with respect to several general representations just accommodated to the specific dataset that is under analysis.

In addition, for a fair comparison, we replicate the training/testing split of all methods versus which we compare and, also we employed publicly available ground truth annotations and *size*, *edges* and *texture* features[4] (see Section 4.4.2) [Rya+15].

In our framework, we set $m = 3$ and each category of features is thus encoded with a separate (quadratic-polynomial or linear) kernel. We fix $c = [1/3, 1/3, 1/3]^\top$ and M^α is the sum of between-view operator from [Min+13] and normalized graph Laplacian related to the α-th view. The model parameters T, λ, γ and $\Delta\xi$ are chosen via cross validation on the training set and, ultimately, is chosen accordingly to the values in Table 4.6.

	Table 4.7		Table 4.9 & 4.8		Table 4.10
	UCSD	*MALL*	*UCSD*	*MALL*	*PETS 2009*
T	0	3	4	3	3
λ	10^{-4}	10^{-4}	10^{-10}	10^{-5}	10^{-5}
γ	10^{-5}	10^{-5}	10^{-11}	10^{-6}	10^{-6}
$\Delta\xi$	0.10	0.15	0.05	0.10	0.10

TABLE 4.6: Number of refinements T, regularizing parameter λ, γ and rate $\Delta\xi$ used by HLR for crowd counting.

For the quantitative performance evaluation, we use the MAE, MSE and MRE metrics (Section 4.3.3) which computes mean absolute, mean squared and mean relative errors between true and estimated number of pedestrians per frame, respectively.

Comparison against [Rya+15]. In Table 4.7, Huber Loss Regression (HLR) is compared with Gaussian Process Regression (GPR), regularized linear regression (Lin), K-nearest neighbors (K-nn) with $K = 1, 2, 4, 8, 16, 32$ and neural networks (NN) methods with a unique hidden layer composed by 4, 8, 16 or

[4] http://personal.ie.cuhk.edu.hk/~ccloy/downloads_mall_dataset.html for *MALL*; http://visal.cs.cityu.edu.hk/downloads/ for *UCSD* and *PETS 2009*.

32 units. Using same training/testing splits as in [Rya+15], we scored top three error on *MALL*, providing the lowest MAE and MRE on *UCSD*.

Method	*UCSD*		*MALL*	
	MAE	MRE	MAE	MRE
GPR	**1.46(2)**	**6.23(2)**	**2.58(1)**	**8.34(1)**
Lin	**1.56(3)**	**6.48(3)**	**2.58(1)**	**8.52(2)**
1-nn	2.89(6)	10.75(8)	3.45(8)	11.22(8)
2-nn	2.77(5)	10.20(5)	3.05(6)	9.94(6)
4-nn	2.72(4)	9.63(4)	2.89(4)	9.28(5)
8-nn	2.90(7)	10.20(5)	2.92(5)	9.20(4)
16-nn	3.12(8)	10.56(7)	3.25(7)	9.96(7)
32-nn	3.76(9)	12.37(9)	4.19(9)	12.51(9)
NN(4)	8.13(11)	33.08(11)	26.06(13)	87.83(13)
NN(8)	9.15(12)	43.08(12)	13.02(11)	43.40(11)
NN(16)	4.36(10)	19.26(10)	16.70(12)	57.61(12)
NN(32)	11.70(13)	54.86(13)	12.18(10)	40.32(10)
HLR	**1.23(1)**	**5.43 (1)**	**2.63 (3)**	**8.74 (3)**

TABLE 4.7: Comparison using the protocol of [Rya+15]. Beside nearest neighbors and neural network, the respective value of K and the number of neurons in the hidden layer are reported. Top three lowest errors in bold, relative ranking in brackets.

Comparison with fully supervised approaches. In Table 4.8), we compare compare HLR on *MALL* and *UCSD* by considering the following approaches as competitors. The least square support vector regression LSSVR of [Ges+01], the kernel ridge regression KRR of [An+07], the random forest regression RFR of [LW02], the Gaussian process regression GPR of [Cha+08], the ridge regression RR of [Sau+98] with its cumulative attribute CA-RR variant of [Che+13a]. Additionally, we also compare with multiple localized regression MLR [Wu+06] and multiple output regression MORR [Che+12a], a class of local approaches for crowd counting which rely on a preliminary fine tessellation of the video frames. Also in this demanding comparison HLR is able to set the lowest and second lowest MAE, MSE and MRE in all of the comparisons, respectively.

Comparison with semi-supervised approaches. Additionally, we benchmarked HLR with other methods which leverage on semi-supervision: namely, the baseline one-viewed manifold regularization (MR) [Bel+06], semi-supervised-regression (SSR) [Loy+13b] and elastic net (EN) [Tan+11b]. SSR optimizes a similar functional to (4.4), where the quadratic loss is used in a multi-view setting ($m = 2$) as to impose a spatial and temporal regularization within- and across-consecutive frames, respectively. EN [Tan+11b] implements a sparsity principle while adopting a L^1-based semi-supervised variation of Lasso. In Table 4.9 we report the MSE quantitative results, where HLR is able to outperform other semi-supervised methods.

Method	*UCSD*			*MALL*		
	MAE	MSE	MRE	MAE	MSE	MRE
LSSVR	2.20(4)	7.69(4)	**0.107(3)**	3.51(4)	18.20(5)	0.108(4)
KRR	**2.16(3)**	**7.45(3)**	**0.107(3)**	3.51(4)	18.18(4)	0.108(4)
RFR	2.42(8)	8.47(8)	0.116(8)	3.91(9)	21.50(8)	0.121(9)
GPR	2.24(5)	7.97(6)	0.112(7)	3.72(7)	20.10(7)	0.115(7)
RR	2.25(6)	7.82(5)	0.110(6)	3.59(6)	19.00(6)	0.110(6)
CA-RR	**2.07(2)**	**6.86(2)**	**0.102(2)**	**3.43(3)**	**17.70(3)**	**0.105(3)**
MLR	2.60(9)	10.10(9)	0.125(9)	3.90(8)	23.90(9)	0.120(8)
MORR	2.29 (7)	8.08(7)	0.109(5)	**3.15(1)**	**15.70(1)**	**0.099(1)**
HLR	**1.99 (1)**	**6.00(1)**	**0.093(1)**	**3.36(2)**	**16.42(2)**	**0.104(2)**

Table 4.8: Comparison of HLR using the protocol of [Che+12a] and [Che+13a]. Top three performance in bold, relative ranking in brackets.

Method	*UCSD*	*MALL*
MR[Bel+06]	7.94(4)	18.42(3)
SSR[Loy+13b]	7.06(3)	17.85(2)
EN[Tan+11b]	6.15(2)	-
HLR	**6.00(1)**	**16.42(1)**

Table 4.9: Comparison with semi-supervised approaches. MSE error metric was used, relative ranking in brackets.

Experiments on *PETS 2009*. Moving to *PETS 2009*, we mimed the protocol of [Cha+09a]. Motion segmentation allows to divide the right-moving pedestrians from the others moving in the opposite direction. Total crowd density has been obtained summing the partial results. Table 4.10 shows a comparison of Huber loss vs. Gaussian Process Regression (GPR). Performances are sometimes substantially improved, see sequence $13-57$, regions R1 and R2. Again, HLR scored a sound performance, setting in 46 cases out of 54 the lowest MAE or MSE error metrics.

Discussion

In comparison with the semi-supervised methods in Table 4.9, the considered multi-view and manifold regularized framework provides a better performance and Huber loss attests to be superior to both the quadratic (MR and SSR) and the L^1 losses (EN). Additionally, the performance with respect to fully supervised method is frequently superior (Tables 4.8 and 4.7), even if using much less annotations.

The auto-unlabeling component is able to proficiently rule the amount of supervision. Indeed, on *UCSD* only 1% of the labels is not exploited by HLR: evidently, the preprocessing step perspective correction [Cha+08] is enough effective to make almost all the data exploitable in a supervised fashion. Differently, on *MALL*, about 11% of labeled instances are discarded: this happens when some pedestrians are partially occluded by some static elements of the scene and, sometimes, there are some sitting people whose appearance greatly differs from the walking ones. Finally, on *PETS 2009*, HLR outperforms GPR

Seq.	Reg.	Method	total		right-moving		left-moving	
			MAE	MSE	MAE	MSE	MAE	MSE
13-57	R0	GPR	2.308	8.362	0.249	0.339	2.475	8.955
		HLR	**2.290**	**8.118**	**0.204**	**0.204**	**2.385**	**8.719**
13-57	R1	GPR	1.697	5.000	0.100	0.100	1.643	4.720
		HLR	**1.330**	**3.005**	**0.059**	**0.059**	**1.290**	**2.919**
13-57	R2	GPR	1.072	1.796	0.235	0.317	0.842	1.484
		HLR	**0.819**	**1.253**	**0.081**	**0.081**	**0.756**	**1.190**
13-59	R0	GPR	1.647	4.087	1.668	4.158	0.154	0.154
		HLR	**1.560**	**3.320**	**1.639**	**3.589**	**0.137**	**0.137**
13-59	R1	GPR	0.685	1.116	0.589	0.871	**0.095**	**0.095**
		HLR	**0.622**	**0.855**	**0.481**	**0.689**	0.166	0.166
13-59	R2	GPR	1.282	**2.577**	1.291	2.436	**0.066**	**0.066**
		HLR	**1.253**	2.747	**1.195**	**2.274**	0.141	0.141
14-06	R1	GPR	4.328	44.159	4.338	44.159	0.005	0.005
		HLR	**4.299**	**43.383**	**4.299**	**43.383**	**0.000**	**0.000**
14-06	R2	GPR	3.139	26.035	3.144	26.129	0.020	0.020
		HLR	**2.995**	**23.970**	**3.015**	**24.289**	**0.020**	**0.020**
14-17	R1	GPR	0.604	1.220	0.604	**1.198**	**0.000**	**0.000**
		HLR	**0.593**	**1.209**	**0.593**	1.209	**0.000**	**0.000**

TABLE 4.10: Comparison of HLR with the protocol of [Cha+09a] on *PETS 2009*. The lowest error is in bold. Sometimes a 0.000 error value is registered: it correspond to the absence of people moving in the specified direction in the given sequence.

even if using, on average, more than 100 annotations less. Despite using less labeled data than competitors, HLR scores a superior performance on *UCSD*, *MALL* and *PETS 2009* datasets.

In terms of running time, the HLR is a fast method: indeed, in the setup of Table 4.8, training and testing on *MALL* last 6.5 and 0.4 seconds respectively. Similarly, on *UCSD*, training requires 5.6 and testing 0.5 seconds.

In synthesis, the crowd counting application showed that HLR is able to fully take advantage of the most effective techniques in semi-supervision and to improve state-of-the-art methods, while being robust to noisy annotations, ensuring a fast computation and skipping annoying parameter-tuning processes.

4.5 Conclusions

In this Chapter we investigated the perspective of combining multiple multimodal data representation and capturing their mutual correlation in order to ultimately allows from a semi-supervised framework which can handle partial supervision within the training set. We observed that, when the available annotations are either noisy or wrong, issues may arise with respect to the procedure of imposing manifold regularity constraints in order to predict unannotated instances on the base of the most closed labelled ones.

In order to tackle this problem, we proposed a novel robust semi-supervised pipeline which exploits the Huber loss function to generate a novel and automatic criterion to inspect the available annotations and remove those which are discovered to be outliers. Such approach leverages on the possibility of automatically learning from the data the threshold function ξ which regulates the Huber loss and which, actually, can be interpreted as the maximum absolute error paid within the labeled part of the training set. We propose to guide the data in order to fix the maximal amount of tolerable noise, just removing those annotations which are prominently not consistent with the average level of reconstruction results that is guaranteed by the majority of the data.

The previous step brings an additional yet remarkable aspect of the proposed approach: we are able to achieve an exact closed-form optimization for the Huber loss, in spite of the very general manifold regularized scalar multi-view regression framework we considered. Interestingly, the automatic label inspection component and the exact optimization are intertwined in one unique module which is iteratively repeated for T times in order to refine the model (usually, T is very small and never exceeds 4 in the experiments we showed hereby). The possibility of exact optimization for the Huber loss for each element of a sequence of thresholds learnt in a data driven manner is very unique and diverse with respect to currently available approaches in the literature [MM00; AZ05; LLZ11; Kha+13] which just optimize the same loss in an approximated sense for an heuristically fixed threshold.

In terms of computational cost, T linear systems need to be solved as to produce the final regression model, therefore allowing for an efficient pipeline. In fact, as already stressed, all components of our approach (manifold regularization, multi-view learning, automatic label inspection, data driven tuning of ξ) are efficiently combined within one unique pipeline so that the aforementioned computational cost takes into account all of them.

While evaluating performance, our approach is able to favorably score in a variety of different setting, including binary classification with randomly flipped labels, classical machine learning regression problems on UCI benchmarks and crowd counting experiments on UCSD, MALL and PETS 2009 datasets. In all cases, our proposed approaches score a favorable performance and, even if our method has been generically devised for scalar regression tasks, it is able to frequently outperform approaches which have been explicitly tailored for only one of the previous applications.

Chapter 5

Unsupervised Deep Domain Adaptation with Geodesic Correlation Alignment and Minimal Entropy

Supervision is a well established paradigm in machine learning and pattern recognition. When the visual categories and concept that one want to recognize are thoroughly annotated, the learner can exploit such wellness of information to build a model which not only is able to perform well on the data used to deploy it but, also, can guarantee a satisfactory generalization capabilities on unknown testing instances, up to some good implementations practice such as regularization. Actually, one of the main reason for the deep learning revolution to have occurred during the last years is the availability of a gigantic corpus of labelled data (especially, images) which can allow for architectures with millions of parameters to be trained – thanks to parallelization and acceleration on GPU – which have now overcome even human capabilities in fine-grained object categorization.

However, the supervision pipeline is a very expensive data regime since annotating data is always time consuming, frequently expensive and often prone to errors since performed by humans. In addition, if comparing with the way humans learn, one may see the asymmetry related to the fact that human cognition is totally fine in learning in unsupervised regimes by capturing the correlations between unknown visual categories and already learnt concepts. That is, by leveraging on a controlled supervised setting where some visual categories are fully annotated and described, human cognition is able to extrapolate from them cues and patterns that turn to be useful in categorizing similar but different concepts in a fully unsupervised fashion. As one example, even if a child is shown a picture of a giraffe, he is totally fine in recognizing real giraffes in a zoo, even if the picture was a bi-dimensional representation

acquired in controlled light setting and on a white background, being those conditions arguably different from the environmental settings of the zoo.

In computer vision, the latter problem can be framed as *unsupervised domain adaptation*. That is, we assume that we are been given a fully annotated dataset on which supervised training can be accomplish in order to distinguish between K distinct visual categories. This labeled set of data is referred as source domain $\mathcal{S}$. The problem is that classification needs to be accomplished not on $\mathcal{S}$ directly but on a different dataset, here called domain, $\mathcal{T}$ that encodes the same K categories in mutated visual setting related to, say, different points of view, illumination changes and background clutter. Globally the latter ambiguities are referred as *domain shift* [TE11]. Since the same K categories are represented in either $\mathcal{S}$ or $\mathcal{T}$, the overall task related to unsupervised domain adaptation is the possibility of overcoming the domain shift issue and adapt the feature representation on the target in such a way that the classifier trained on $\mathcal{S}$ can be applied to those while achieving a solid performance which does not suffer of performance degradation related to the shift between domains.

In the recent past years, a broad class of approaches has leveraged on *entropy optimization* as a proxy for (unsupervised) domain adaptation, borrowing the idea from semi-supervised learning [GB04]. By either performing entropy regularization [Tze+15a; Car+17; Sai+17], explicit entropy minimization [Hae+17c; Hae+17b], or implicit entropy maximization through adversarial training [GL15; Tze+17], this statistical tool has demonstrated to be a powerful technique for domain adaptation.

Actually, optimizing the entropy can be interpreted as an indirect approach to learn a transformation that aligns the source and the target domain statistics. Among the methods which directly seek for such transformation, correlation alignment minimizes the distance between second order statistics, in order to make the source data distribution more similar to the target one [Sun+16; SS16].

Apparently, correlation alignment and entropy minimization may seem two alternative approaches in optimizing models for domain adaptation. However, in this Chapter, we will show that this is not the case and, indeed, we claim that the two classes of approaches are deeply intertwined. We formally demonstrate this claim, and at the same time, we also obtain a solution for the prickly problem of hyperparameter validation in unsupervised domain adaptation[1].

In summary, this Chapter brings the following contributions.

1. We explore the two paradigms of correlation alignment and entropy minimization, by formally demonstrating that, at its optimum, correlation alignment attains the minimum of the sum of cross-entropy on the source domain and of the entropy on the target.

2. Motivated by the urgency of penalizing correlation misalignments in practical terms, we observe that an Euclidean penalty, as adopted in [Sun+16;

[1]The hyperparameter problem arises since, by supposing that no labels from the target set are available *at all*, one can construct a validation set out of *source* data only, which is not helpful since not representative of *target* data. Note that if *some* labels from the target set were available, the problem would be cast into *semi-supervised domain adaptation*, which is out of the scope of our work.

SS16], is not taking into account the structure of the manifold where covariance matrices lie in. We thus propose a different loss function that is inspired by a geodesic distance that takes into account the manifold's curvature while computing distances.

3. When aligning second order statistics, a hyper-parameter controls the balance between the reduction of the domain shift and the supervised classification on the source domain. In this respect, a manual cross-validation of the parameter is not straightforward: doing it on the source domain may not be representative, and it is not possible to do on the target due to the lack of annotations. Owing to our principled connection between correlation alignment and entropy regularization, we devise an entropy-based criterion to accomplish such validation in a data-driven fashion.

4. We combine the geodesic correlation alignment with the entropy-based criterion in a unique pipeline that we call *minimal-entropy correlation alignment*. Through an extensive experimental analysis on publicly available benchmarks for transfer object categorization, we certify the effectiveness of the proposed approach in terms of systematic improvements over former alignment methods and state-of-the-art techniques for unsupervised domain adaptation in general.

5.1 Euclidean correlation alignment (CORAL)

We describe our method by taking a multi-class classification problem. Suppose we are given a source domain with N training examples, each of those being a d dimensional vector: we can represent the source domain data in the form of a $N \times d$ matrix $\mathbf{X}$ where samples $\mathbf{x}_i$ are stacked by rows. For each of those, we have with corresponding labels which we stack in a $N \times K$ matrix $\mathbf{Z}$ in which is row is a one-hot-vector encoding in which the k-th component $\mathbf{z}_i(k) = 1$ if the corresponding instance $\mathbf{x}_i$ belongs to class k, being $\mathbf{z}_i(k) = 0$ otherwise. Similarly, we are given *unannotated* target data $\mathbf{x}'_j \in \mathbb{R}^d$, $j = 1, \dots, M$ which we similarly stack by rows in a $M \times d$ matrix $\mathbf{X}'$. We can now compute the mean feature embeddings $\boldsymbol{\mu}_{\mathbf{X}}$ and $\boldsymbol{\mu}_{\mathbf{X}'}$ relative to either the source or the target according to

$$\begin{aligned} \boldsymbol{\mu}_{\mathbf{X}} &= \frac{1}{N} \sum_{i=1}^{N} \mathbf{x}_i \\ \boldsymbol{\mu}_{\mathbf{X}'} &= \frac{1}{M} \sum_{j=1}^{M} \mathbf{x}'_j. \end{aligned} \tag{5.1}$$

Through (5.1), we can compute the source and target covariance representations $\mathbf{C}_{\mathbf{X}}$ and $\mathbf{C}_{\mathbf{X}'}$ whose generic (p, q) entries, $p, q = 1, \dots, d$ is computed according

to

$$\mathbf{C}_X(p,q) = \frac{1}{N-1}\sum_{i=1}^{N}(\mathbf{x}_i(p)-\boldsymbol{\mu}_X(p))(\mathbf{x}_i(q)-\boldsymbol{\mu}_X(q))$$
$$\mathbf{C}_U(p,q) = \frac{1}{M-1}\sum_{j=1}^{M}(\mathbf{x}'_j(p)-\boldsymbol{\mu}_{X'}(p))(\mathbf{x}'_j(q)-\boldsymbol{\mu}_{X'}(q)). \tag{5.2}$$

To minimize the distance between the second-order statistics (covariance) of the source and target features, one may apply a linear transformation $\mathbf{A}$ to the original source features and use the Frobenius norm $\|\cdot\|_F$ as the matrix distance metric. Thus, one gets

$$\min_{\mathbf{A}\in\mathbb{R}^{d\times d}} \|\mathbf{A}\mathbf{C}_X\mathbf{A}^\top - \mathbf{C}_{X'}\|_F^2. \tag{5.3}$$

If $\text{rank}(\mathbf{C}_X) \geq \text{rank}(\mathbf{C}_U)$, then an analytical solution can be obtained by choosing $\mathbf{A}$ to be the identity matrix. However, the data typically lie on a lower dimensional manifold, so the covariance matrices are likely to be low rank [Cai+10]. For such particular case, one can still solve the problem by means of the following result.

Theorem 7. *Let* $\mathbf{V}\boldsymbol{\Sigma}\mathbf{V}^\top = \mathbf{C}_X$ *and* $\mathbf{V}'\boldsymbol{\Sigma}'\mathbf{V}'^\top = \mathbf{C}_X$ *the eigendecompositions of* $\mathbf{C}_X$ *and* $\mathbf{C}_{X'}$*, respectively. Further, let us denote* r *as the minimum between the two ranks of* $\mathbf{C}_X$ *and* $\mathbf{C}_{X'}$ *and let us denote* $\mathbf{V}'_{[:,1:r]}$ *the matrix obtained by selecting the* r *columns of* $\mathbf{V}'$ *corresponding to the* r *largest eigenvalues of* $\mathbf{C}_{X'}$ *which are collected in the* $r\times r$ *diagonal matrix* $\boldsymbol{\Sigma}'_{[1:r,1:r]}$*. Then*

$$\mathbf{A}^{\text{opt}} = \mathbf{V}\boldsymbol{\Sigma}^{-1/2}\mathbf{V}^\top\mathbf{V}'_{[:,1:r]}\boldsymbol{\Sigma}'^{1/2}_{[1:r,1:r]}\mathbf{V}'^\top_{[:,1:r]} \tag{5.4}$$

is the global minimizer of (5.3)*.*

Proof. See [Sun+16] □

The previous result inspires the CORAL algorithm which is visualized in Figure 5.1. We can think of the transformation $\mathbf{A}$ in (5.3) intuitively as follows: $\mathbf{V}\boldsymbol{\Sigma}^{-1/2}\mathbf{V}^\top$ whitens the source data, while $\mathbf{V}'_{[:,1:r]}\boldsymbol{\Sigma}'^{1/2}_{[1:r,1:r]}\mathbf{V}'^\top_{[:,1:r]}$ re-colors it with the target covariance. This is illustrated in Figure 5.1 - middle and bottom, respectively. Globally, the correlation alignment pipeline as proposed in [Sun+16] is formally presented through the following pseudo-code.

Algorithm 6: CORAL

Input : Source data $\mathbf{X}$, target data $\mathbf{X}'$
Output: Aligned source data $\mathbf{X}^\circ$

1 Compute $\mathbf{C}_X$ and $\mathbf{C}_{X'}$ as in (5.2);
2 Compute the eigenvectors $\mathbf{V}$ and $\boldsymbol{\Sigma}$ of $\mathbf{C}_X$;
3 Whiten the source $\overline{\mathbf{X}} = \mathbf{X}\mathbf{V}\boldsymbol{\Sigma}^{-1/2}\mathbf{V}^\top$;
4 Compute the eigenvectors $\mathbf{V}'$ and $\boldsymbol{\Sigma}'$ of $\mathbf{C}_{X'}$;
5 Re-coloring the source using the target $\mathbf{X}^\circ = \overline{\mathbf{X}}\mathbf{V}'\boldsymbol{\Sigma}'^{1/2}\mathbf{V}'^\top$;

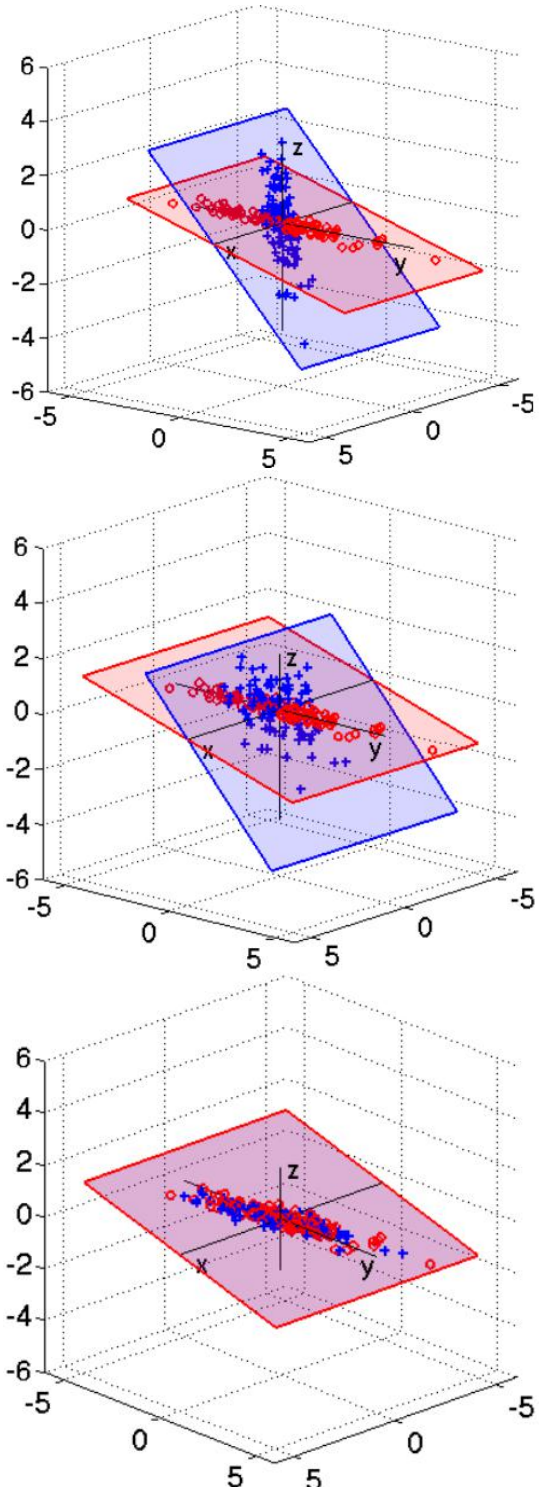


FIGURE 5.1: Illustration of CORAL. *Top*. The original source and target domains have different distribution covariances, despite the features being normalized to zero mean and unit standard deviation. This presents a problem for transferring classifiers trained on source to target. *Middle*. The same two domains after source decorrelation in which we remove the feature correlations of the source domain. *Bottom*. Target re-correlation, adding the correlation of the target domain to the source features. After this step, the source and target distributions are well aligned and the classifier trained on the adjusted source domain is expected to work well in the target domain. Image and caption courtesy of [Sun+16]. Best viewed in colors.

5.2 Geodesic correlation alignment

Deep Neural networks are able to learn powerful hierarchical representations from large training sets and show very good generalization capabilities. Furthermore, learned features are so general that can often be successfully transferred across domains and tasks, especially when one has the possibility to fine tune the network exploiting labelled examples from the new domain. Yet, deep architectures are not completely immune from the so called *domain shift* problem [TE11], i.e. they suffer from performance degradation under changes in the input data distribution [Don+14]. This is what typically happens when training supervised machine learning algorithms to be deployed in "the wild", where test (*target*) distributions can be extremely different from the training one (*source*), and labeled test/validation samples are usually not available. This issue is known under the name of *domain adaptation* (DA) and addressing it, especially in the unsupervised case, is in fact critical for successfully applying machine learning to real-world applications.

Actually, one would like to avoid collecting and labeling data to train a new classifier for any new possible scenario, and in some real cases this is not actually possible. Instead, it would be desirable to find methods that cope with the degradation in classification performance by effectively transferring the knowledge acquired on the labeled source domain to some unlabeled target domain [Fer+13].

To cope with this issue, a possibility consists in minimizing the distance between source and target data marginal distributions so as to estimate a transformation of the feature representations which may lead to better classification. In other words, this is also equivalent to *confuse* the domains so that a classifier cannot distinguish between source and target domains.

Many DA works tackle this problem according to this intuition and propose methods aimed at *aligning* information extracted from the domains' data, be either lower dimensional manifolds, subspaces or distributions [Fer+13; Sun+16] This alignment can be performed in an unsupervised fashion, although it can strongly benefit from some labeled samples [Tze+15b].

In this context, CORAL (Section 5.1) is a "frustratingly easy" unsupervised domain adaptation method which finds the linear transformation which minimizes the Frobenius norm of the difference between the covariance matrices of source and target data features.

Although straighforward, the method looks effective, however, it shows two main drawbacks. First, it relies on a linear transformation, which could be insufficient to capture the most appropriate feature transformations. In fact, the way features are (cor)related across the domains is not known, and assuming a linear relations is big bet, especially if deep feature representations (e.g., from last fully connected layers, *fc7* and/or *fc8*) are considered. Second, it is not end-to-end since it needs to extract features from both source and target datasets, calculate covariances, align the features, and then train a classifier.

Indeed, the latter drawback of CORAL was recently addressed by Deep CORAL [SS16], which incorporates the alignment of second-order statistics into a deep

architecture, by proposing a loss term which minimizes the batch-wise difference between source and target data correlations. In essence, Deep CORAL aims at optimizing the weights of a deep architecture jointly considering the optimization problem of the covariance difference and the standard classification problem. This is done by designing a loss composed by the standard (cross-entropy) classification loss and another, properly weighted, loss penalizing the covariances' difference.

We operate just in this context by addressing a fundamental issue inherent of the above approaches. Although CORAL and Dep CORAL showed good results and proved effective, they overlooked fundamental properties of covariances, which turn out to be Symmetric Positive Definite (SPD) matrices. A key property of such matrices is that, given $n \in \mathbb{N}$, the set of $n \times n$ SPD matrices is not a subspace of the Euclidean space, but instead has the structure of a Riemannian manifold with non-positive curvature, usually denoted as $\mathrm{Sym}^{++}(n)$. As a consequence, methods for manipulating elements in $\mathrm{Sym}^{++}(n)$ which simply rely on the Euclidean metric are usually *suboptimal* [Min+14b; Min+16b; Cav+16; Cav+17c; Cav+17a]. This is quite intuitive, since the Frobenius norm of a matrix difference, $\|A - B\|_F$, is defined only in terms of the element-wise difference $A - B$, without reflecting any structure in A and B.

Many methods have been proposed in the literature which exploit the non-Euclidean nature of $\mathrm{Sym}^{++}(n)$ [Dry+09].

A common approach exploits the affine-invariant metric, a classical Riemannian metric on $\mathrm{Sym}^{++}(n)$ [CP12; Pen+06], which is typically computationally intensive, particularly when large-scale applications are faced. Another approach exploits Bregman divergences on $\mathrm{Sym}^{++}(n)$ [Kul+06; Che+13b]. These are not Riemannian metrics but are quite fast to compute and proved to work properly on retrieval tasks, even considering very simple classification methods such as nearest-neighbor methods. The computational complexity is undoubtely one of the main drawback affecting the manipulation of such objects which limits the usage of such tools, despite their elegant and rigorous mathematical soundness. This motivated the development of the *Log-Euclidean metric* framework [Ars+07; Wan+12e], which is faster than the affine-invariant metric and, moreover, is a Riemannian metric on $\mathrm{Sym}^{++}(n)$ (unlike the Bregman divergences), and thus can better suits its manifold structural form.

The latter distance was recently exploited in computer vision and machine learning tasks [Min+16b; Cav+16] since it is fast to compute, and in particular, has the interesting property of being differentiable. In our case, this property permits to compute gradients with respect to each entry of the source and target covariance matrices allowing end-to-end optimization via gradient descent techniques.

In this Section, we show that, leveraging the Riemannian structure of $\mathrm{Sym}^{++}(n)$, domain adaptation can be performed in a more effective and principled way. In fact, second order statistics must be properly aligned within their natural embedding manifold, instead of naively projected in the Euclidean space. Since the Euclidean distance is proved to be suboptimal on the curved manifold $\mathrm{Sym}^{++}(n)$, we introduce a loss function based on the Log-Euclidean metric, introducing a novel, more rigorous version of the Deep CORAL framework. This allows to effectively and correctly align second order statistics of the

source and target data domains, whose effect results even more evident whenever source and target datasets are characterized by very different marginal distributions. Experiments performed on a standard domain adaptation benchmark, the Office dataset, show superior performance, empirically confirming the correctness of our approach. As a side finding, we notice a pathological behavior of the Euclidean distance, which can be interpreted as a symptom of its sub-optimality with respect to geodesic distances in $\mathrm{Sym}^{++}(n)$.

5.2.1 Background: covariance matrices and manifold distances

CORAL (CORrelation ALignment) [Sun+16] is an unsupervised DA method which consists in aligning second-order statistics of source and target data distributions (typically, after normalization and zero-mean transformations). It finds the linear transformation A which minimizes the Frobenius norm $\|\cdot\|_F$ of the difference between the covariance matrices of source and target data, C_S and C_T respectively, by solving:

$$\min_A \|C_{\hat{S}} - C_T\|_F^2 = \min_A \|A^T C_S A - C_T\|_F^2. \tag{5.5}$$

The transformation A^* which, acting on the source data, minimizes (5.5), has essentially the form of a whitening operation, followed by a re-coloring of the whitened features, performed through the covariance operator of the target domain. It can be applied in any kind of DA problem, regardless of the chosen features (and also deep features can be used) and classification method used afterwards.

The minimization problem (5.5) is analytically solved to provide the optimal solution [Sun+16]:

$$A^* = (U_S \Sigma_S^{+\frac{1}{2}} U_S^T)(U_{T[1:r]} \Sigma_{T[1:r]}^{\frac{1}{2}} U_{T[1:r]}^T). \tag{5.6}$$

Here r is the minimum rank of the source and target covariance, $r = \min(r_{C_s}, r_{C_T})$, $U_S \Sigma_S U_S^T$ is the diagonalization of C_S, Σ_S^+ is the Moore-Penrose pseudoinverse of Σ and $U_{T[1:r]} \Sigma_{T[1:r]} U_{T[1:r]}^T$ are the largest r singular values and corresponding vectors resulting from the diagonalization of C_T . A^* is cleary made by a first part, which whitens the source data and a second one which re-colors it with the target statistics. However, in practice, for the sake of efficiency and stability, CORAL employs the standard whitening and recoloring, where a small regularization term $\gamma \mathbb{I}_d$ is added to the covariance matrices in order to make them explicitly full rank and positive definite.

Deep CORAL [SS16] incorporates the alignment of second-order statistics into a deep architecture, by proposing a loss term which minimizes the batch-wise difference between source and target correlations. Deep CORAL indeed aims at optimizing the weights of a deep architecture by jointly solving the problem (5.5) and the standard classification problem, designing by a compound loss so composed:

$$L = L_{CLASS} + \lambda L_{CORAL}, \tag{5.7}$$

where

$$L_{CORAL} = \frac{1}{4d^2} \|C_S - C_T\|_F^2. \tag{5.8}$$

and L_{CLASS} is the standard cross-entropy loss. L_{CORAL} is calculated (for each batch) over the d-dimensional *fc8* features of AlexNet [Kri+12a], but according to [SS16] could possibly include contributions from hidden representations at each level of the network. This formulation permits to compute gradients with respect to each entry of the source and target covariance matrices allowing end-to-end optimization via gradient descent techniques.

5.2.2 Geodesic alignment

Covariance representations are SPD matrices which live in a Riemannian space $Sym^{++}(n)$, and metrics defined therein should take into account its non-Euclidean structure, so the (Euclidean) distance present in (5.8) is only suboptimal in such a space. The *Log-Euclidean metric* is instead a Riemannian metric and better captures the manifold structure. It is characterized by some interesting properties (which will be illustrated in the following) and is defined as:

$$d_{logE}(X, Y) = \| \log(X) - \log(Y) \|_F, \tag{5.9}$$

where $\log(A)$ is defined as $\log(A) = U \mathrm{diag}(\log(\lambda_1), ..., \log(\lambda_n)) U^T$ through the spectral decomposition $A = U \mathrm{diag}(\lambda_1, ..., \lambda_n) U^T$.

In the context of CORAL, it is straightforward to note that the solution of problem (5.5), A^*, incidentally minimizes the analogous problem, that is:

$$\min_A \| \log(C_{\hat{S}}) - \log(C_T) \|_F^2 = \min_A \| \log(A^T C_S A) - \log(C_T) \|_F^2. \tag{5.10}$$

Actually, supposing to work with the regularized full-rank matrices (which is indeed what people do in practice), the minimization problem (5.5) finds the transformation A^* which makes the Frobenius distance *null*, i.e., the transformation which realizes the equality $C_{\hat{S}} = C_T$. This of course also yields $\log(C_{\hat{S}}) = \log(C_T)$, which minimizes eq. (5.10) as a consequence of the fact both metrics are well defined (in either spaces, they comply with distance properties), and satisfy $d(X, Y) = 0 \Leftrightarrow X = Y$. Namely, what matters in CORAL is only the analytical function which directly transforms the data to make the two covariances *the same*.

But the Deep CORAL [SS16] methods works quite differently. In this case, domain adaptation is end-to-end, and the problem is addressed by jointly optimizing the loss for the supervised problem and the Euclidean loss in eq. (5.8). The transformation A^* is now implicitly learned step-by-step by the network via gradient descent. In other words, the final deep features are both discriminative enough to train a strong classifier and invariant (to some extent) to the difference between source and target domains. However, the minimization is now a smooth process, which follows a precise path in the parameter (weight) space. Such a path naturally induces a trajectory in $Sym^{++}(n)$ which connects the covariance of the source with the one of the target. It is thus natural to constrain such a path to be a *geodesic* trajectory, enforcing a minimum distance which takes into account the curvature of the manifold $Sym^{++}(n)$.

As a side consideration, let us note that a *perfect* alignment of the source and target distributions up to second order statistics is indeed a very strong assumption done in CORAL [Sun+16]. A more reasonable and milder constraint

is to have a balance between good features in the source domain and a sound statistical adaptation to the target distribution.

The LOG-D-CORAL loss. Based on the above considerations, we propose to address the unsupervised domain adaptation problem by adding to a deep network a loss term based on a geodesic distance on $\mathrm{Sym}^{++}(n)$, namely the log-Euclidean one, which, as already mentioned, offers theoretical and practical advantages over other metrics on $\mathrm{Sym}^{++}(n)$.

$$L_{log} = \frac{1}{4d^2}\| \log(C_S) - \log(C_T)\|_F^2 \tag{5.11}$$

where d is the dimension of the hidden features whose covariances are intended to be aligned, U and V are the matrices which diagonalize C_S and C_T, respectively, and λ_i and μ_i, $i = 1, ..., d$ are the corresponding eigenvalues. The normalization term $1/d^2$ accounts for the sum of the d^2 terms in the Frobenius distance, which makes the loss independent from the size of the features.

Jointly training with a standard classification loss and the proposed loss in (5.11) allows to learn features which do not overfit the source data since they reflect the statistical structure of the target set. Hence, the total loss reads

$$L = L_{CLASS} + \alpha L_{log}. \tag{5.12}$$

The hyperparameter α is a critical coefficient. A high value of α is likely to force the network towards learning oversimplified low-rank feature representations, which may have perfectly aligned covariances but would be useless for classification purposes. On the other hand, a small α may not be enough to fill the domain shift.

Differentiability. The loss (5.11) needs to be differentiable in order for the minimization problem to be solved via back-propagation, and its gradients should be calculated with respect to the input features. Given a zero mean data matrix $D \in \mathbb{R}^{L\times d}$, composed by L samples of d dimensional vectors, its covariance is simply proportional to the quadratic form D^TD, whose gradients can be straightforwardly computed.

The scenario is indeed more complicated than expected since the logarithm of an SPD matrix is defined through its eigendecomposition in eq. (5.11). Fortunately, eigenvalues and eigenvectors are differentiable functions for SPD matrices [KM03]. Lastly, the point-wise log is applied in (5.11) on the matrix listing *strictly positive* eigenvalues[2] on the diagonal, and it is thus differentiable everywhere as a function of λ_i and μ_i.

In practice, modern tools for deep learning consist in software libraries for numerical computation whose core abstraction is represented by *computational graphs*. Single mathematical operations (e.g., matrix multiplication, summation etc.) are deployed on nodes of a graph and data flows through edges. Reverse-mode differentiation [Gri12] takes advantage of the gradients of single operations, allowing training by backpropagation through the graph [Ola]. The loss

[2] Remember that covariances are in practice regularized by adding a small perturbation $\gamma\mathbb{I}$.

(5.11) can be easily written in few lines of code by exploiting mathematical operations already implemented, together with their gradients, in TensorFlow™[Tf].

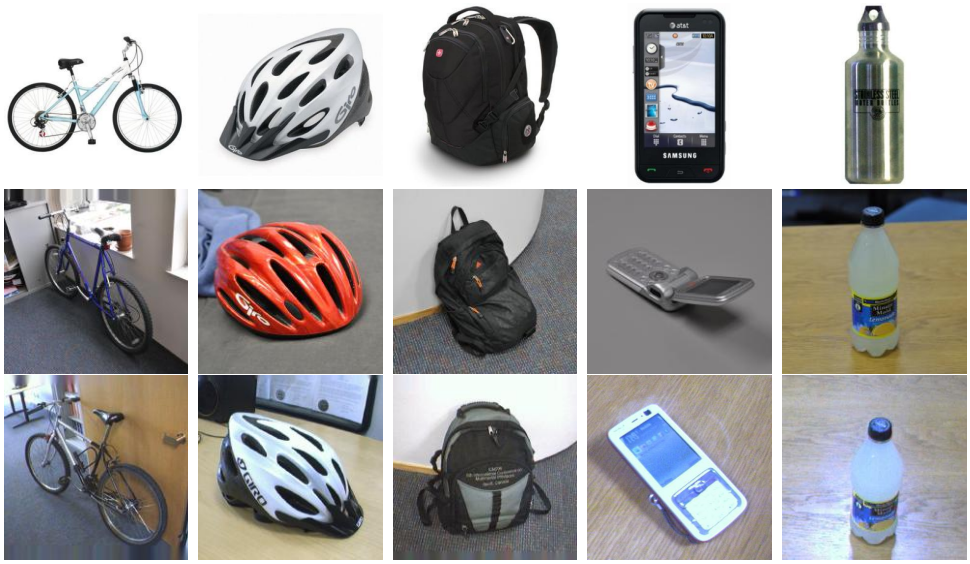


FIGURE 5.2: Samples from 5 classes of the Office [Sae+10] dataset. Each row correspond to one of the three different domains, namely Amazon, DSLR, Webcam (top to bottom). Similarity between DSLR and AMAZON reflects in very high recognition accuracies in the evaluation setups W→D and D→W, as reported in Table 5.1. In fact, the effect of domain adaptation techniques are in general more evident on the the other domain shifts.

We reproduce the experimental setup proposed in Deep Coral [SS16], by validating our approach on a the standard domain adaptation benchmark - the Office dataset [Sae+10]. The dataset consists in images belonging to 31 different object categories, gathered from 3 different domains, namely *Amazon*, *DSLR* and *Webcam* (Figure 5.2). The most standard evaluation protocol for unsupervised domain adaptation (see e.g.[Gon+12; Tze+14; GL15]) is simple: given the three domains, there are 6 possible domain shifts. Training is performed on the the fully labeled data from a given source domain, while testing is done over the two remaining (targets). Only *unlabeled* data from the current target domain is available at training time, which allows to work out statistics to fill the domain gap.

Implementation details. We fine tune AlexNet [Kri+12a] pre-trained on Imagenet [Den+09], by setting the dimension of its last hidden layer (*fc8*) to 31, i.e. the number of classes in the Office dataset. The new layer is initialized with Gaussian noise $\mathcal{N}(0, 5 \times 10^3)$, batch size is 128 and base learning rate 10^{-3}, with scheduled exponential decreasing. Batches are made of both target and source examples, where the former contribute to L_{log} or L_{CORAL} losses only, while the latter (which are labeled) also contribute to the cross entropy loss L_{CLASS}. The network is trained separately on the three domains and tested on the remaining ones, to serve as a baseline (first row of Table 5.1). Unfortunately we were not able to reproduce the performance of AlexNet reported by [SS16] by some percentage points, although we accurately followed all of

its prescriptions. This does not really matter since we are only interested in the *relative gain* in performance introduced by our loss function with respect to the suboptimal Euclidean one. Loss weights α and λ used for each domain shift are listed in Table 5.2, while the covariance regularizer γ was set to 10^{-5} once for all. The implementation is in TensorFlow™[Tf] and our Python code will be made publicly available.

	A→D	A→W	D→A	D→W	W→A	W→D
AlexNet [Kri+12a]	57.8	60.2	40.0	95.2	40.0	97.8
Deep Coral [SS16]	58.9	65.9	**40.7**	**95.6**	**41.6**	98.0
Log D-Coral	**62.0**	**68.5**	40.6	95.3	40.6	**98.7**

	Average *(gain)*
AlexNet [Kri+12a]	58.6
Deep Coral [SS16]	60.5 *(+1.9)*
Log D-Coral	**61.4** ***(+2.8)***

TABLE 5.1: Object recognition accuracies (percentage) for the 6 standard splits of the Office dataset. Each split represents a domain shift SOURCE → TARGET.

Discussion Percentage classification accuracies are reported in Table 5.1. The average percentage gain of Deep CORAL is consistent with the 2% gain published in [SS16]. The log loss introduced in this work contributes with an additional 1% approximately. Our approach achieves better accuracies in 3 out of the 6 splits. However the margins are very small in the remaining cases, with respect to both the baseline and Deep CORAL.

The results of Table 5.1 prove i) that covariance alignment is indeed effective in filling the domain gap ii) that covariances must be regarded as matrices belonging to their natural embedding space, i.e. $\mathrm{Sym}^{++}(n)$, and should thus be compared with appropriate distance measures.

In order to get a better understanding of the difference between Deep CORAL and Log-D-Coral we plot in Figure 5.3 their weighted losses αL_{log} and λL_{CORAL} (as from equations (5.8) and (5.11) for the domain shift A → W.

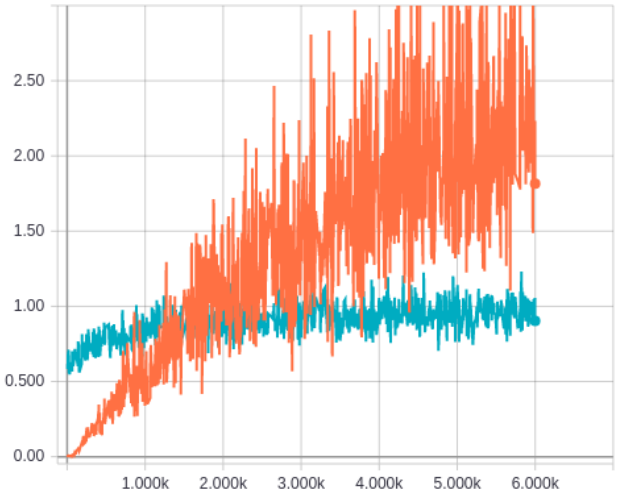
Loss weight	A→D	A→W	D→A	D→W	W→A	W→D
λ *(Deep Coral [SS16])*	1.	1.	0.05	0.05	0.1	0.1
α *(Log D-Coral)*	10.	10.	0.1	0.1	5.	1.

TABLE 5.2: Hyperparameters wighting the covariance losses. They had to be chosen differently for each domain shift, since each represents an independent problem, where domain adaptation is needed to a different extent. We found that, in general α has to be chosen approximately one order of magnitude higher than λ.

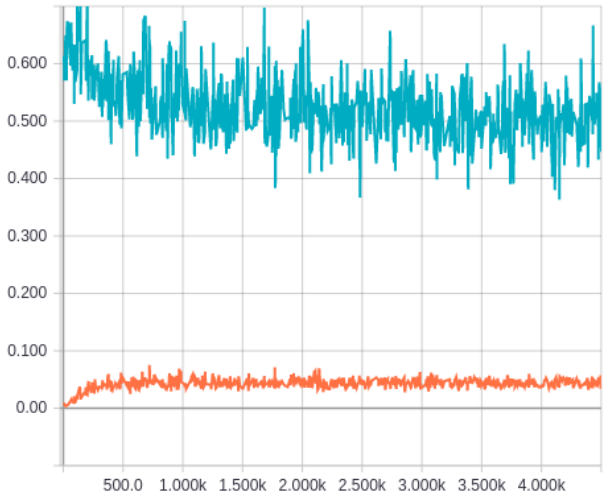
Figure 5.3(a) shows the batch-wise value of the two distance terms in a normal training, i.e. the two losses are calculated but no domain adaptation is

enforced. Both losses naturally increase since the statistics of feature representations learned from the source are likely to diverge from the target ones, as the network specializes more and more to source data. However, L_{CORAL} shows a pathological behavior, still increasing at an almost linear rate even when training reaches convergence and weights are only slightly updated. One would expect that little variations in the weights should instead produce little variations in the distance. On the contrary, L_{log}, even though increasing as expected, shows a way more reasonable trend. In fact, as training approaches convergence, L_{log} tends to stabilize. Last, L_{CORAL} is more noisy than L_{log} (oscillation are bigger despite we plot λL_{CORAL}, with $\lambda \ll \alpha$), meaning that its value can change a lot from one batch to another, which is a very undesirable property, meaning that the batch-wise distance is not well representative of the distance between the whole source and target datasets. This behavior can be interpreted as evidence of the sub-optimal nature of the Euclidean metric with respect to geodesic distances in $Sym^{++}(n)$.

Figure 5.3(b) depicts instead the value of the two losses included in the minimization problem. As reported in [SS16], L_{CORAL} experiences stabilization after increasing for a few epochs. This behavior is quite unclear, since we are trying to minimize it, but possibly depends on the base value of the distance on the uninitialized network. L_{log}, on the contrary, stabilizes within few epochs, after being minimized, which is somehow more reasonable. Oscillations are here comparable given that we plot αL_{log} and λL_{CORAL}, with $\alpha = 10\lambda$).



(a)



(b)

FIGURE 5.3: Batch-wise value of the terms αL_{log} (cyan) and λL_{CORAL} (orange) for the A→W split. (a) A standard training (*no domain adaptation*), i.e. no distance constraint is enforced in the total loss function. (b) *Domain adaptation*: L_{log} and L_{CORAL} are here added to the loss function and minimized jointly with the classification loss.

5.3 Geodesic alignment with joint entropy minimization

In this Section, we will detail the two classes of correlation alignment and entropy optimization methods that are combined by our adaptation technique.

Background and problem formulation. We consider the problem of classifying an image $\mathbf{x}$ in a K-classes problem. To do so, we exploit a bunch of labeled images $\mathbf{x}_1, \ldots, \mathbf{x}_n$ and we seek for training a statistical classifier that, during inference, provides probabilities for a given test image $\mathbf{x}$ to belong to each of the K classes. In this work, such classifier is fixed to be a deep multi-layer

feed-forward neural network denoted as

$$f(\mathbf{x};\theta) = [\mathbb{P}(\text{class}(\mathbf{x}) = 1), \quad \mathbb{P}(\text{class}(\mathbf{x}) = 2), \quad \dots, \quad \mathbb{P}(\text{class}(\mathbf{x}) = K)]. \tag{5.13}$$

The network f depends upon some parameters/weights θ that are optimized by minimizing over θ the cross-entropy loss function

$$H(\mathbf{X}, \mathbf{Z}) = -\sum_{i=1}^{n} \langle \mathbf{z}_i, \log f(\mathbf{x}_i;\theta) \rangle. \tag{5.14}$$

In (5.14), for each image $\mathbf{x}_i$, the inner product $\langle \cdot, \cdot \rangle$ computes a similarity measure between the network prediction $f(\mathbf{x}_i;\theta)$ and the corresponding data label $\mathbf{z}_i$, which is a K dimensional one-hot encoding vector. Precisely, $z_{ik} = 1$ if $\mathbf{x}_i$ belongs to the k-th class, being zero otherwise. Finally, for notational simplicity, let $\mathbf{X}$ and $\mathbf{Z}$ define the collection all images $\mathbf{x}_i$ and corresponding labels $\mathbf{z}_i$, respectively.

In a classical fully supervised setting, other than minimizing (5.14), one can also add some weighted additive regularizers to the final loss, such as an L^2 penalty. But, in the case of domain adaptation, θ should be chosen as to promote a good portability from the source $\mathcal{S}$ to the target domain $\mathcal{T}$.

Correlation alignment. In the case of unsupervised domain adaptation, we assume that none of the examples in the target domain is labelled and, therefore, we should perform adaptation at the feature level. In the case of correlation alignment, we can replace (5.14) with the following problem

$$\min_{\theta} \left[H(\mathbf{X}_\mathcal{S}, \mathbf{Z}_\mathcal{S}) + \lambda \cdot \ell(\mathbf{C}_\mathcal{S}, \mathbf{C}_\mathcal{T}) \right], \qquad \lambda > 0, \tag{5.15}$$

where we compute the supervised cross-entropy loss between data $\mathbf{X}_\mathcal{S}$ and annotations $\mathbf{Z}_\mathcal{S}$ belonging to the source domain only. Concurrently, the network parameters θ are modified in order to align the covariance representations

$$\mathbf{C}_\mathcal{S} = \mathbf{A}_\mathcal{S} \mathbf{J} \mathbf{A}_\mathcal{S}^\top, \quad \text{and} \quad \mathbf{C}_\mathcal{T} = \mathbf{A}_\mathcal{T} \mathbf{J} \mathbf{A}_\mathcal{T}^\top \tag{5.16}$$

that are computed through the centering matrix $\mathbf{J}$ (see [Min+14b; Cav+16] for a closed-form) on top of the activations computed at a given layer[3] by the network $f(\cdot, \theta)$. Precisely, $\mathbf{A}_\mathcal{S}$ and $\mathbf{A}_\mathcal{T}$ stack by columns the d-dimensional activations computed from the source and the target domains. Also, θ is regularized according to the following Euclidean penalization

$$\ell(\mathbf{C}_\mathcal{S}, \mathbf{C}_\mathcal{T}) = \frac{1}{4d^2} \|\mathbf{C}_\mathcal{S} - \mathbf{C}_\mathcal{T}\|_F^2 \tag{5.17}$$

in terms of the (squared) Frobenius norm $\|\cdot\|_F$. In [Fer+13; Sun+16], the aligning transformation is obtained in closed-form. Despite the latter would attain the perfect correlation matching, it requires matrix inversion and eigen-decomposition operations: thus it is not scalable. As a remedy, in [SS16], (5.17) is used a loss for optimizing (5.15) with stochastic batch-wise gradient descent.

[3] In principle, correlation alignment can be done at multiple layers in parallel, but empirical evidences [Sun+16; SS16] suggest that a solid performance is achieved even if it's done only once.

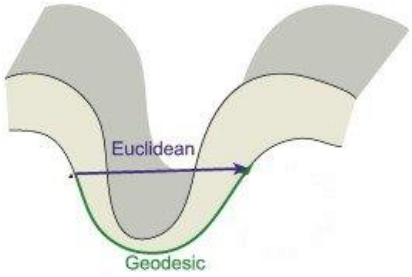


FIGURE 5.4: Geodesic versus Euclidean distances in the case of a non-zero curvature manifold (as the one of SPD matrices).

Problem 1. *Mathematically, covariance representations* (5.16) *are symmetric and positive definite (SPD) matrices belonging to a Riemannian manifold with non-zero curvature [Ars+07]. Therefore, measuring correlation (mis)alignments with an Euclidean metric like* (5.17) *is arguably suboptimal since it does not capture the inner geometry of the data (see Figure 5.4).*

Entropy regularization. The cross entropy H on the source domain and entropy E on the target domain can be optimized as follows:

$$\min_{\theta} \left[H(\mathbf{X}_{\mathcal{S}}, \mathbf{Z}_{\mathcal{S}}) + \gamma E(\mathbf{X}_{\mathcal{T}}) \right], \quad \gamma > 0 \tag{5.18}$$

where

$$E(\mathbf{X}_{\mathcal{T}}) = -\sum_{\mathbf{x}_t \in \mathcal{T}} \langle f(\mathbf{x}_t; \theta), \log f(\mathbf{x}_t; \theta) \rangle. \tag{5.19}$$

In this way, we circumvent the impossibility of optimizing the cross entropy on the target (due to the unavailability of labels on $\mathcal{T}$), and we replace it with the entropy $E(\mathbf{X}_{\mathcal{T}})$ computed on the soft-labels $\mathbf{z}_{\text{soft}}(\mathbf{x}_t) = f(\mathbf{x}_t; \theta)$, which is nothing but the network predictions [Lee13]. Empirically, soft-labels increases the confidence of the model related to its prediction. However, for the purpose of domain adaptation, optimizing (5.18) is *not enough* and, in parallel, ancillary adaptation techniques are invoked. Specifically, either additional supervision [Tze+15a], batch normalization [Car+17] or probabilistic walk on the data manifold [Hae+17c; Hae+17b] have been exploited. As a different setup, a min-max problem can be devised where $H(\mathbf{X}_{\mathcal{S}}, \mathbf{Z}_{\mathcal{S}})$ is minimized and, at the same time, entropy is maximized within a binary classification of predicting whether a given instance belongs to the source or the target domain. This is done in [GL15] and [Tze+17] by reversing the gradients and using adversarial training, respectively. In practical terms, this means that, in addition to the loss function in (5.18), one needs to carry out other parallel optimizations whose reciprocal balance in influencing the parameters' update is controlled by means of hyper-parameters. Since the latter have to be grid-searched, a validation set is needed in order to select the hyper-parameters' configuration that corresponds to the best performance on it. How to select the aforementioned validation set leads to the following point.

Problem 2. *In the case of domain adaptation, cross-validation for hyper-parameter tuning on the source directly is unreasonable because of the domain shift. In*

fact, for instance, [Tze+15a] can do it only by adding supervision on the target and, in [Car+17], cross-validation is performed on the source after *the target has been aligned to it. Since we need* λ *to be fixed before solving for correlation alignment and since we consider a fully unsupervised adaptation setting, we cannot use any of the previous strategy and, obviously, we are not allowed for supervised cross-validation on the target. Thus, hyper-parameter tuning is really a problem.*

In this work, we combine the two classes of correlation alignment [Sun+16; Fer+13; SS16] and entropy optimization [Tze+15a; GL15; Tze+17; Hae+17c; Hae+17b; Car+17] in a unique framework. By doing so, we embrace a more principled approach to align covariance representations (as to tackle Problem 1), while, at the same time, solving Problem 2 with a novel unsupervised and data-driven cross-validation technique.

5.3.1 Optimal Correlation Alignment Induces Entropy Minimization

In this section, we deploy a rigorous mathematical connection between correlation alignment and entropy minimization in order to understand the mutual relationships. The following theorem represents the main result.

Theorem 8. *With the notation introduced so far, if* $\theta^\star$ *optimally aligns correlation in* (5.15)*, then,* $\theta^\star$ *minimizes* (5.18) *for every* $\gamma > 0$.

Proof. By hypothesis, we assume that $\theta^\star$ is the optimal hyper-parameter which attains the optimum of (5.15), which implies

$$H(\mathbf{X}_\mathcal{S}, \mathbf{Z}_\mathcal{S}) = \min \quad \text{and} \quad \mathbf{C}_\mathcal{S} = \mathbf{C}_\mathcal{T}, \tag{5.20}$$

by the properties of the squared-distance function d.

Let us fix an arbitrary $\gamma > 0$ and let us consider

$$L(\theta) = H(\mathbf{X}_\mathcal{S}, \mathbf{Z}_\mathcal{S}) + \gamma E(\mathbf{X}_\mathcal{T}). \tag{5.21}$$

the objective functional in (5.18) which rewrites

$$L(\theta) = -\sum_{\mathbf{x}_i \in \mathcal{S}} \log \left(\sum_{k=1}^{K} z_{ik} f_k(\mathbf{x}_i; \theta) \right) - \gamma \sum_{\mathbf{x}_j \in \mathcal{T}} \sum_{k=1}^{K} f_k(\mathbf{x}_j; \theta) \log \left(f_k(\mathbf{x}_j; \theta) \right) \tag{5.22}$$

while writing down the expression of the cross-entropy function H between ground truth source labels $Z_\mathcal{S}$ and network's predictions which are also exploited to compute the entropy function E on the target domain.

By hypothesis, since $\theta^\star$ is such that $H(\mathbf{X}_\mathcal{S}, \mathbf{Z}_\mathcal{S}) = \min$, then the thesis will follow if we prove that

$$E(\mathbf{X}_\mathcal{T}) = -\gamma \sum_{\mathbf{x}_j \in \mathcal{T}} \sum_{k=1}^{K} f_k(\mathbf{x}_j; \theta^\star) \log \left(f_k(\mathbf{x}_j; \theta^\star) \right) = \min \tag{5.23}$$

since the minimum of the sum of two functions is achieved when the two addends are minimized separately. Now, by hypothesis, since we assume the optimal correlation alignment, then, due to the fact that $\mathbf{C}_{\mathcal{S}} = \mathbf{C}_{\mathcal{T}}$, we can assume that the statistical properties of the trained classifier on the source can be transferred to the target with null performance degradation since, basically, we have obtained the way to completely solve the domain shift issue. This implies that, if we assume that some oracle will provide us the ground truth labels $\mathbf{z}_j$ for the target domain, we can get that

$$f(\mathbf{x}_j; \theta^\star) = \mathbf{z}_j \tag{5.24}$$

for any arbitrary $\mathbf{x}_j$ in the target domain $\mathcal{T}$. Note that $\theta^\star$ was optimized in a fair manner, by exploiting the labels of the source domain only and the fact that a perfect classification on the target is achieved is a side effect of assuming that we achieved the optimal correlation alignment, making the target data distribution essentially indistinguishable from the source one. In particular, $f(\mathbf{x}_j; \theta^\star)$ is a Dirac's delta function such that $f_k(\mathbf{x}_j; \theta^\star) = 1$ if $\mathbf{x}_j$ belongs to the k-th class and $f_k(\mathbf{x}_j; \theta^\star) = 0$ otherwise. Therefore, we get

$$-\gamma \sum_{\mathbf{x}_j \in \mathcal{T}} \sum_{k=1}^{K} f_k(\mathbf{x}_j; \theta^\star) \log\left(f_k(\mathbf{x}_j; \theta^\star)\right) = -\gamma \sum_{\mathbf{x}_j \in \mathcal{T}} \left[\sum_{k \neq \text{class}(\mathbf{x}_j)} 0 + \log 1 \right] \tag{5.25}$$

due to the fact that ${}_k(\mathbf{x}_j; \theta^\star)$ is a Dirac's delta and since we decompose, for each $\mathbf{x}_j$, the summation over k in two parts: when k equals the class of $\mathbf{x}_j$, $f_k(\mathbf{x}_j; \theta^\star) \log\left(f_k(\mathbf{x}_j; \theta^\star)\right) = \log 1 = 0$ and, in all other cases, the addends vanishes. Therefore

$$-\gamma \sum_{\mathbf{x}_j \in \mathcal{T}} \sum_{k=1}^{K} f_k(\mathbf{x}_j; \theta^\star) \log\left(f_k(\mathbf{x}_j; \theta^\star)\right) = 0. \tag{5.26}$$

Since $E(\mathbf{X}_{\mathcal{T}})$ is a non-negative function, (5.26) gives the thesis (5.23) due to the generality of γ. □

The previous statement certifies that, at its optimum, correlation alignment provides minimal entropy for free. If one compares (5.15) with (5.18), one may notice that, in both cases, we are minimizing H over the source domain $\mathcal{S}$. Therefore, if we assume that $H(\mathbf{X}_{\mathcal{S}}, \mathbf{Z}_{\mathcal{S}}) = \min$, we have a perfect classifier whose predictions on $\mathcal{S}$ are extremely confident and correct. Thus, the predictions are distributed in a very picky manner and, therefore, entropy on the source is minimized. At the same time, we can minimize the entropy on the target since $\mathcal{T}$ is made "indistinguishable" from $\mathcal{S}$ after the alignment. Hence, the target's predictions are distributed in a similar picky way so that entropy on $\mathcal{T}$ is minimized as well.

Observation 1. *Since we proved that optimal correlation alignment implies entropy minimization, one may ask whether the converse holds. That is, if the optimum of* (5.18) *gives the optimum of* (5.15)*. The answer is negative as it will be clear by the following counterexample. In fact, we can always minimize the cross entropy on the source with a fully supervised training on* $\mathcal{S}$*. However, such classifier could be always confident in classifying a target example as belonging to, say, Class* 1*. After that, we can deploy a dummy adaptation step that, for*

whatever target image $\mathbf{x}$ *to be classified, we always predict it to be Class 1. In this case the entropy on the target is clearly minimized since the distribution of the target prediction is a Dirac's delta* δ_{1k} *for any class* k*. But, obviously, nothing has been done for the sake of adaptation and, in particular, optimal correlation alignment is far from being realized.*

In fact, consider the fully supervised classification problem of optimizing θ *for the deep neural network* $f(\cdot, \theta)$ *such that, while comparing network's prediction* $f(\mathbf{x}_i, \theta)$ *and ground truth annotations* $\mathbf{z}_i$, relative to the source domain $\mathcal{S}$*, we get the following problem.*

$$\textit{Train the network } f = f(\,\cdot\,; \theta^\star) \textit{ by solving for } \theta^\star = \arg\min_\theta H(\mathbf{X}_\mathcal{S}, \mathbf{Z}_\mathcal{S}). \tag{5.27}$$

Now, we can devise a dummy classifier $\widetilde{f}$*, depending upon the same exact parameter choice* θ *such that*

$$\widetilde{f}(\mathbf{x}; \theta) = \begin{cases} f(\mathbf{x}; \theta) & \textit{if } \mathbf{x} \in \mathcal{S} \\ [1, 0, \dots, 0] & \textit{if } \mathbf{x} \in \mathcal{T}. \end{cases} \tag{5.28}$$

That is, we use on the target the same exact classifier that we trained on the source (with no adaptation). That is, source data is classified by $\widetilde{f}$ *based on* f*, while, when asked to classify an image from the target domain,* $\widetilde{f}$ *will always predict that instance to belong to the first class. By using the same exact scheme of proof as in Theorem 8, we can show that,* $\widetilde{f}$ *achieves the minimal entropy* $E(\mathbf{X}_\mathcal{T})$ *on the target domain* $\mathcal{T}$*. This is an evidence for the fact that, although optimal correlation alignment implies minimal entropy, the converse is not true. Ancillary, it explains why in [Tze+15a; Car+17], adaptation is effectively carried out with ancillary techniques and entropy regularization it's just a boosting factor as opposed to a factual regularizer for domain adaptation.*

In Theorem 8, the assumption of having an optimal correlation alignment is crucial for our theoretical analysis. However, in practical terms, optimal alignment is also desirable in order to effectively deploy domain adaptation systems. Moreover, despite the optimal alignment in (5.15) is able to minimize (5.18) for any $\gamma > 0$, in practice, hyper-parameters need to be cross-validated and this is not an easy task in unsupervised domain adaptation (as we explained in Problem 2). In the next section, a solution for all these problems will be distilled from our improved knowledge.

5.3.2 Minimal-Entropy Correlation Alignment (MECA)

Based on the previous remarks, we address the unsupervised domain adaptation problem by training a deep net for supervised classification on $\mathcal{S}$ while adding a loss term based on a geodesic distance on the SPD manifold. Precisely, we consider the (squared) log-Euclidean distance

$$\begin{aligned} \ell_{\log}(\mathbf{C}_\mathcal{S}, \mathbf{C}_\mathcal{T}) &= \|\log(\mathbf{C}_\mathcal{S}) - \log(\mathbf{C}_\mathcal{S})\|_F^2 \\ &= \frac{1}{4d^2} \Big\| \mathbf{U}\operatorname{diag}(\log(\sigma_1), \dots, \log(\sigma_d))\mathbf{U}^\top + \\ &\qquad -\mathbf{V}\operatorname{diag}(\log(\mu_1), \dots, \log(\mu_d))\mathbf{V}^\top \Big\|_F^2 \end{aligned} \tag{5.29}$$

where d is the dimension of the activations $\mathbf{A}_{\mathcal{S}}$ and $\mathbf{A}_{\mathcal{T}}$, whose covariances are intended to be aligned, $\mathbf{U}$ and $\mathbf{V}$ are the matrices which diagonalize $\mathbf{C}_{\mathcal{S}}$ and $\mathbf{C}_{\mathcal{T}}$, respectively, and σ_i, μ_i, $i = 1, ..., d$ are the corresponding eigenvalues. The normalization term $1/d^2$ accounts for the sum of the d^2 terms in the $\|\cdot\|_F^2$ norm, which makes $\ell_{\log}$ independent from the size of the feature layer.

The geodesic alignment for correlation is attained by minimizing the problem $\min_\theta [H(\mathbf{X}_{\mathcal{S}}, \mathbf{Z}_{\mathcal{S}}) + \lambda \cdot \ell_{\log}(\mathbf{C}_{\mathcal{S}}, \mathbf{C}_{\mathcal{T}})]$, for some $\lambda > 0$. This allows lo learn good features for classification which, at the same time, do not overfit the source data since they reflect the statistical structure of the target set. To this end, a geodesic distance accounts for the geometrical structure of covariance matrices better than (5.15). In this respect, the following two aspects are crucial.

- *Problem 1* is addressed by introducing the log-Euclidean distance $\ell_{\log}$ between SPD matrices, which is a geodesic distance widely adopted in computer vision [Cav+16; Zha+16a; Min+14b; Min+16b; Cav+17a] when dealing with covariance operators. The rationale is that, within the many geodesic distances, (5.29) is extremely efficient because does not require matrix inversions (like the affine one $\ell_{\text{aff}}(\mathbf{C}_{\mathcal{S}}, \mathbf{C}_{\mathcal{T}}) = \|\log(\mathbf{C}_{\mathcal{S}}\mathbf{C}_{\mathcal{T}}^{-1})\|_F$). Moreover, while shifting from one geodesic distance to another, the gap in performance obtained are negligible, provided the soundness of the metric [Zha+16a].
- As observed in *Problem 2*, the hyperparameter λ is a critical coefficient to be cross validated. In fact, a high value of λ is likely to force the network towards learning oversimplified low-rank feature representations. Despite this may result in perfectly aligned covariances, it could be useless for classification purposes. On the other hand, a small λ may not be enough to bridge the domain shift. Motivated by Theorem 8, we select the λ which minimizes the entropy $E(\mathbf{X}_{\mathcal{T}})$ on the target domain. Indeed, since we proved that $H(\mathbf{X}_{\mathcal{S}})$ is minimized at the same time in both (5.15) and (5.18), we can naturally tune λ so that $E(\mathbf{X}_{\mathcal{T}}) = \min$. Note that this entropy-based criterion for λ is totally fair in unsupervised domain adaptation since, as in (5.18), E does not require ground truth target labels to be computed, but only relies on inferred soft-labels.

In summary, we propose the following minimization pipeline for unsupervised domain adaptation, which we name *Minimal-Entropy Correlation Alignment* (*MECA*)

$$\boxed{\min_\theta [H(\mathbf{X}_{\mathcal{S}}, \mathbf{Z}_{\mathcal{S}}) + \lambda \cdot \ell_{\log}(\mathbf{C}_{\mathcal{S}}, \mathbf{C}_{\mathcal{T}})] \text{ subject to } \lambda \text{ minimizes } E(\mathbf{X}_{\mathcal{T}}).} \tag{5.30}$$

In other words, in (5.30), we minimize the objective functional $H(\mathbf{X}_{\mathcal{S}}, \mathbf{Z}_{\mathcal{S}}) + \lambda \cdot \ell_{\log}(\mathbf{C}_{\mathcal{S}}, \mathbf{C}_{\mathcal{T}})$ by gradient descent over θ. While doing so, we can choose λ by validation, such that the network $f(\cdot; \theta)$ is able, at the same time, to attain the minimal entropy on the target domain.

Differentiability. For a fixed λ, the loss (5.30) needs to be differentiable in order for the minimization problem to be solved via back-propagation, and its gradients should be calculated with respect to the input features. However, as (5.16) shows, $\mathbf{C}_{\mathcal{S}}$ and $\mathbf{C}_{\mathcal{T}}$ are polynomial functions of the activations and the same holds when one applies the Euclidean norm $\|\cdot\|_F^2$. Additionally, since the log function is differentiable over its domain, we can easily see that we

can still write down the gradients of the loss (5.30) in a closed form by exhaustively applying the chain rule over elementary functions that are in turn differentiable. In practice, this is not even needed, since modern tools for deep learning consist in software libraries for numerical computation whose core abstraction is represented by *computational graphs*. Single mathematical operations (e.g., matrix multiplication, summation etc.) are deployed on nodes of a graph, and data flows through edges. Reverse-mode differentiation takes advantage of the gradients of single operations, allowing training by backpropagation through the graph. The loss (5.30) can be easily written (for a fixed λ) in few lines of code by exploiting mathematical operations which are already implemented, together with their gradients, in TensorFlow™ or other libraries. We made our code publicly available at the link `https://github.com/pmorerio/minimal-entropy-correlation-alignment` for evaluation purposes.

Experimental evaluation. We will corroborate our theoretical analysis with a broad validation which certify the correctness of Theorem 8 and the effectiveness of our proposed entropy-based cross-validation for λ in (5.30). In addition, by means of a benchmark comparison with state-of-the-art approaches in unsupervised domain adaptation, we will prove the effectiveness of the geodesic versus the Euclidean alignment and, in general, that MECA outperforms many previously proposed methods.

We run the following adaptation experiments. We use digits from SVHN [Net+11] as source and we transfer on MNIST. Similarly, we transfer from SYN DIGITS [GL15] to SVHN. For the object recognition task, we train a model to classify objects on RGB images from NYUD [Sil+12] dataset and we test on (different) depth images from the same visual categories. We also considered the Office dataset [Sae+10] and the related object recognition challenge. Detailed presentations of datasets and technical details for reproducibility follow.

SVHN $\rightarrow$ MNIST. This split represents a very realistic domain shift, since SVHN [Net+11] (Street-View-House-Numbers) is built with real-world house numbers. We used the whole training sets of both datasets, following the usual protocol for unsupervised domain adaptation (SVHN's training set contains $73,257$ images). We also resized MNIST images to 32×32 pixels and converted SVHN to grayscale, according to the standard protocol.

The architecture employed is the very same employed in [GL15] with the only difference that the last fully connected layer (`fc2`) has only 64 units instead of 2048. Performances are the same, but covariance computation is less onerous. `fc2` is in fact the layer where domain adaptation i performed.

Office. The dataset consists in images belonging to 31 different object categories, gathered from 3 different domains, namely *Amazon*, *DSLR* and *Webcam* (Figure 5.2). The most standard evaluation protocol for unsupervised domain adaptation (see e.g.[Gon+12; Tze+14; GL15]) is simple: given the three domains, there are 6 possible domain shifts. Training is performed on the the fully labeled data from a given source domain, while testing is done over the two remaining (targets). Only *unlabeled* data from the current target domain

FIGURE 5.5: Sampled images from the datasets involved in the domain adaptation experiments. From left to right, SVHN (first column, digits 9, 9, 2 from top to bottom), SYN (second column, digits 3, 9, 7 from top to bottom), NYUD RGB (third column, `toilet`, `sink` and `garbage-bin` classes acquires as RGB), NYUD depth (fourth column, different instances from the same previous classes acquired with the alternative modality) and the well known MNIST dataset (fifth column, from top to bottom, digits 0, 4, 6).

is available at training time, which allows to work out statistics to fill the domain gap.

NYUD (RGB → depth). This domain adaptation problem is actually a *modality adaptation* task and it was recently proposed by Tzeng et al. [Tze+17]. The dataset is gathered by cropping out object bounding boxes around instances of 19 classes of the NYUD [Sil+12] dataset. It comprises 2,186 labeled source (RGB) images and 2,401 unlabeled target depth images, HHA-encoded [Gup+14]. Note that these are obtained from two different splits of the original dataset, in order to ensure that the same instance is not seen in both domains. The adaptation task is extremely challenging, due to the very different nature of the data, the limited number of examples (especially for some classes) and the low resolution anf heterogeneous size of the cropped bounding boxes.

We fine-uned a VGG in order to be comparable with ADDA baseline in [Tze+17]. Covariance alignment occurs at `fc8`, which is replaced with a 64-unit layer.

SYN DIGITS → SVHN. This split represents a synthetic-to-real domain adaptation problem, of great interest for research in computer vision, since often requires less efforts generating labeled synthetic data than obtaining large labeled dataset with real samples. SYN DIGITS [GL15] contains 500,000 images belonging to the same SVHN's classes.

As the baseline network, we used the same model as for **SVHN → MNIST.**, but `fc1` has 3072 units.

Validating our theoretical analysis. As shown in Theorem 8, correlation alignment and entropy regularization are intertwined. Despite this result holds at the optimum only, we can actually observe an even stronger linkage. Precisely, we empirically register that a gradient descent path for correlation alignment induces a gradient descent path for entropy minimization. In fact, in the top-left part of Figure 5.6, while running correlation alignment to align source and target with either an Euclidean (red curve) or geodesic penalty (orange curve), we are able to minimize the entropy. Also, when comparing the two, geodesic provides a lower entropy value than the Euclidean alignment, meaning that our approach is able to better minimize $\mathsf{E}(\mathbf{X}_{\mathcal{T}})$. Interestingly, even if the baseline with no adaptation is able to minimize the entropy as well (blue curve), this is only a matter of overfitting the source. In fact, the baseline produces a classifier which is overconfidently wrong on the target as long as training evolves. Remember that optimal correlation alignment implies entropy minimization being the converse not true: if we check the alignment of source and target distributions (Figure 5.6 bottom-left), we see that, with no adaptation (blue curve), the two distributions are increasingly mismatched as long as training proceeds. Differently, with either Euclidean or geodesic alignments, we are able to match the two and, in order to check the quality of such alignment, we conduct the following experiment.

In Figure 5.6, right column, we show the plots of target entropy and classification accuracies related to SVHN→MNIST as a function of λ, where λ varies in $\{0.1, 0.5, 1, 2, 5, 7, 10, 20\}$. Let us stress that, since we measure distances on the SPD manifold directly, we can conjecture that (5.29) can achieve a better alignment between covariances than (5.17). Actually, if one applies the closed-form solution of [Sun+16] the optimal alignment can be found analytically. However, due to the required matrix inversions, such approach is not scalable an one needs to backpropagate errors starting from a penalty function in order to train the model. As one can clearly see in Figure 5.6 (right), Euclidean alignment is performing about 5% worse than our proposed geodesic alignment on SVHN→MNIST. But, most importantly, in the Euclidean case, the minimal entropy does not correspond to the maximum performance on the target. Differently, when using the geodesic penalty (5.29), we see that the λ which minimizes $\mathsf{E}(\mathbf{X}_{\mathcal{T}})$ is also the one that gives the maximum performance on the target. Thus, we can conclude that our geodesic approach is better than the Euclidean one since totally compatible with a data-driven cross-validation strategy for λ, requiring no labels belonging to the target domain.

Additional evidences of the superiority of our proposed geodesic alignment in favor of a classical one are reported in the next Section. Thereby, our Minimal-Entropy Correlation Alignment (*MECA*) method is benchmarked against state-of-the-art approaches for unsupervised deep domain adaptation.

Improving unsupervised domain adaptation with MECA We benchmark MECA against general state-of-the-art frameworks for unsupervised domain adaptation with deep learning: Domain Separation Network (DSN) [Bou+16] and Domain Transfer Network (DTN) [Tai+17]. In addition, we also compare with two (implicit) entropy maximization frameworks - Gradient Reversal Layer (GRL) [GL15] and ADDA [Tze+17] - and with the entropy regularization

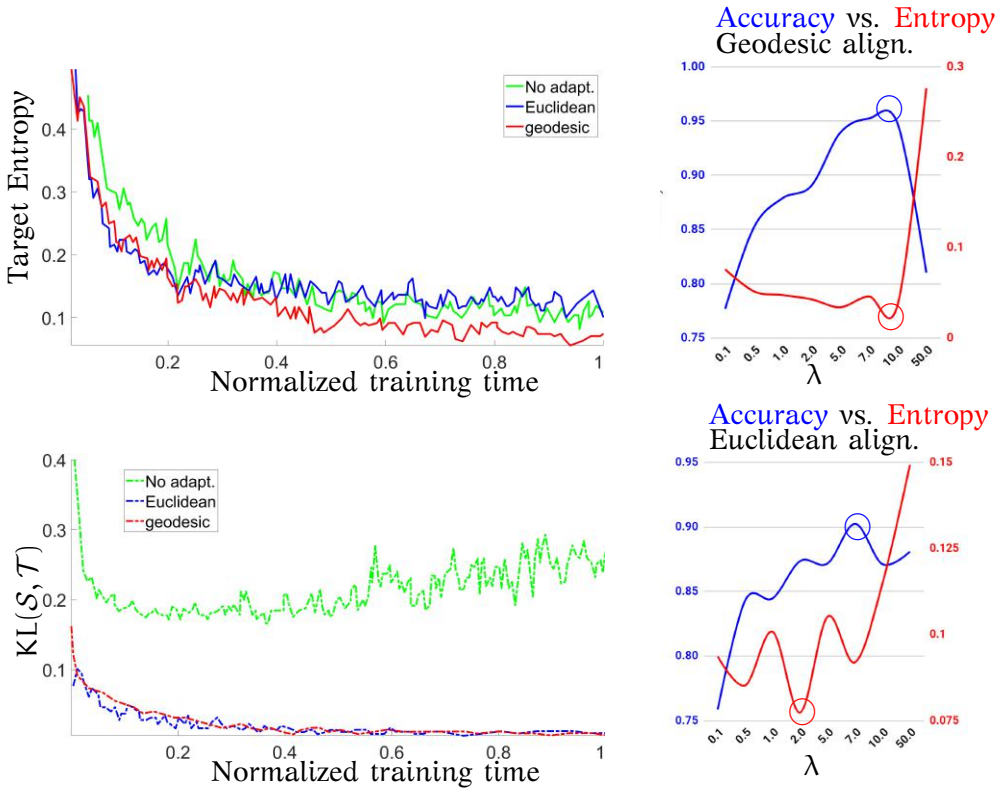


FIGURE 5.6: A gradient descent path for correlation alignment induces a gradient descent path for entropy minimization. *Left column.* We compare a baseline CNN trained on the source (SVHN) only (blue), with the same model where we applied either Euclidean (red) or geodesic alignment (orange) with $\lambda = 0.1$ using MNIST as target. We compare the target entropy (top) and the correlation alignment (bottom) with a KL divergence between source and target distribution. *Right column.* Target accuracy versus target entropy as a function of λ for Euclidean (bottom) or geodesic (top) correlation alignment. Best viewed in colors.

technique of [Sai+17], which uses a *triple* classifier (TRIPLE). Also, we consider the deep Euclidean correlation alignment named Deep CORAL [SS16]. In order to carry on a comparative analysis, we setup standard baseline architectures which reproduce *source only* performances (i.e., performance of the models with no adaptation).

In all cases, we report the published results from the other competitors, even when they devised more favorable experimental conditions than ours (*e.g.*, DTN exploits the extra data provided with SVHN). In the case of Deep CORAL, since the published results only cover the (almost saturated) Office dataset, we decided to run our own implementation of the method. While doing this, in order to cross-validate λ in (5.15), we tried to balance the magnitudes of the two losses (5.14) and (5.17) as prescribed in the original work. However, since this approach does not provide good results, we were forced to cross-validate Deep Coral on the target directly. Let us remark that our proposed entropy-based cross validation is not always compatible with an Euclidean alignment. Differently, for MECA, our geodesic approach naturally embeds the entropy-based criterion and, consequently, we are able to maximize the performance on the target with a fully unsupervised and data-dependent cross-validation.

	A→D	A→W	D→A	D→W	W→A	W→D
AlexNet [Kri+12a]	57.8	60.2	40.0	95.2	40.0	97.8
Deep Coral [SS16]	58.9	65.9	40.7	95.6	**41.6**	98.0
***MECA** (proposed)*	**62.8**	**69.5**	**41.4**	**96.7**	41.2	**98.8**

TABLE 5.3: Object recognition accuracies (percentage) for the 6 standard splits of the Office dataset. Each split represents a domain shift SOURCE → TARGET. With respect to the values reported in Table 5.1, the performance of joint entropy minimization and correlation alignment leads to a superior performance if compared to a simple correlation alignment.

In addition, the classification performance registered by MECA is extremely solid. On the Office dataset, promising results are obtained in the challenging setups (A→D and A→W) where images with a neutral background and uniform viewpoint are used in training, while evaluating the model on more realistic data. Such big domain gap makes the a geodesic correlation alignment extremely effective, if compared with the baseline and Deep Coral. In the other benchmark of the Office dataset, where the gap between source and target is less demanding, the performance of the proposed method are more closed to each others since the adaptation benchmarks are less demanding. With respect to the values reported in Table 5.1, the performance of joint entropy minimization and correlation alignment leads to a superior performance if compared to a simple correlation alignment.

Moving to the digit classification experiments, in the worst case we found (SYN→SVHN), MECA is performing practically on par with respect to Deep CORAL, despite for the latter labels on the target are used, being not far from the score of TRIPLE. This point can be explained with the fact that, for some benchmark datasets, the domain shift is not so prominent - e.g., check the visual similarities between SYN and SVHN datasets in the first two columns of Figure 5.5. In such cases, one can naturally argue that the type of alignment

Method		SVHN→MNIST	NYUD	SYN→SVHN
Source only: baseline		*0.685*	*0.139*	*0.870*
Train on target[§]		*0.994*	*0.468*	*0.922*
DSN [Bou+16]		0.827	-	0.912
DTN[†] [Tai+17]		0.844	-	-
GRL [GL15]	(E)	0.739	-	0.911
ADDA [Tze+17]	(E)	0.760	0.211	-
TRIPLE [Sai+17]	(E)	0.862	-	**0.931**
Deep CORAL[‡] [SS16]	(**C**)	0.902	0.224	0.898
***MECA** (proposed)*	(E + **C**)	***0.952***	***0.255***	*0.903*

Table 5.4: Unsupervised domain adaptation with *MECA*. Perfomance is measured as normalized accuracy and we compare with general, entropy-related (E) and correlation alignment (**C**) state-of-the-art approaches. [§]We also include this experiment exclusively for evaluation purposes. Let us stress that all methods in comparisons and our proposed MECA exploit labels only from the source domain during training. [†]A more powerful feature extractor as baseline and uses also extra SVHN data. [‡]Results refer to our own Tensorflow™ implementation, with cross-validation on the target.

is not so crucial since adaptation is not strictly necessary, and the two types of alignment are pretty equivalent. This also explains the gap shown by MECA from the state-of-the-art (TRIPLE, 93.1%, which performs better than training on target with our architecture) and, eventually, the fact that the baseline itself is already doing pretty well (87.0%). As the results certify, MECA is systematically outperforming Deep CORAL: +0.5% on SYN→SVHN, +2.1% on NYUD and +5% on SVHN→MNIST.

Finally, our proposed MECA is able to improve the previous methods by margin on SVHN→MNIST (+5.0%) and on NYUD as well (+2.6%).

5.4 Conclusion

In this Chapter, we demonstrate that models trained with full supervision on an annotated dataset can be proficiently extended towards different visual domain upon a minor semi-supervised adaptation which does not require any target instance to be annotated. This leads to a broad applicability in real world cases where collecting annotations can be difficult or onerous while, at the same time, totally removing human biases in categorizing objects and classes.

For the latter task we proved that aligning second-order statistics between domains is an effective technique for the purpose of unsupervised domain adaptation. Such approach can be effectively implemented within any existing architecture since acting as an additive regularizer to the final classification loss which is augmented with a term which promotes the alignment between source and target covariance representations (as defined in (5.16)).

Nevertheless, two major issues arise.

Prob. 1 In previously proposed alignment strategies, the intrinsic mathematical structure of the manifold in which covariance representation lie is actually disregarded. In fact, in either [Sun+16] or [SS16], an Euclidean loss is exploited to measure misalignments and this is clearly suboptimal since covariance matrices belong to the SPD manifold which has non-zero curvature.

Prob. 2 Algorithmically, augmenting the classification loss with an ancillary term requires to cross-validate a weight to control the balance between the two. Usually such balancing is obtained via cross-validation, but, in the case of unsupervised domain adaptation, this is not actually an easy task. Indeed, cross validating on a sub-portion of the source is likely to be not-indicative of the target performance due to the already cited domain shift. At the same, time, due to the lack of annotations for the target domain in the full unsupervised setting, usual grid searching schemes are actually not applicable.

For each of those, we proposed a separate solution.

Sol. 1 We adopt a more principled strategy in aligning covariance representation which adopts a geodesic alignment in order to compute (mis)-alignments through geodesics. This is implemented by replacing the Euclidean loss of [Sun+16; SS16] with the log-Euclidean penalty (5.29) which is a Riemanninan metric, thus inducing a geodesic distance, and, a the same time, it's the most efficient with respect to all other geodesic distances, since does not require matrix inversions and due to the fact that (5.29) decouples in the logarithmic terms which can be computed in advance. Moreover, recent findings [Zha+16a] suggest that all Riemannian metrics on the SPD manifold lead to an almost equivalent performance.

Sol. 2 By means of our principled approach, we are able to better achieve alignments and, as we can guarantee after our theoretical analysis - Section 5.3.2 - we are guaranteed to minimize entropy as well. Inspired by the connection we derived, we propose to cross-validate the balance between classification loss (on the source) and source-to-target alignment by considering the setup which best minimizes the target entropy. Since the latter is computed only in terms of pseudo-labels [Lee13], which are nothing but network's predictions, we can still apply this criterion even when no annotation is available in the target domain.

These two components, when combined in our proposed minimal-entropy correlation alignment (MECA), provide an efficient pipeline for unsupervised domain adaptation which ensure a solid performance against state-of-the-art methods in benchmarks object recognition problems.

Chapter 6

Dropout: Counteracting Overfitting by Discouraging Over-Correlations for Representation Learning

Hand crafting descriptors for computer vision is a good practice to encode prior human expertise. However, the generalization capabilities of this class of representations may not be satisfactory. For instance, since wheel has a well defined geometric shape, that is circular and enriched by threads. However, in real world scenarios, the appearance of a wheel may be complicated by either shadows falling on threads (and making them not visible) or from perspective issues which makes the wheel's shape elliptical rather than circular. These kinds of variations are hard to account for manually. Instead, we can let the representation learning neural network learn them from data by giving it several positive and negative examples of a wheel (as well as other visual categories) and training it end-to-end.

With this respect, learning the representation from the data itself is clearly beneficial in order to spot, in an automatic manner, the most salient cues that can be useful. However, a problem arises in this paradigm: it can be the case that the learnt feature representation are excessively correlated in the sense that they are not capturing the most relevant characteristic embedded in the data but, instead, they are memorizing the data. Clearly, the latter case is extremely unfavorable because it leads to a sever drop in performance while shifting from the training set to a unknown test instance and, mainly, it's due because of the variety of free parameters that need to be optimized in modern deep learning architecture. In other words, such over-complicated models lead to overfitting.

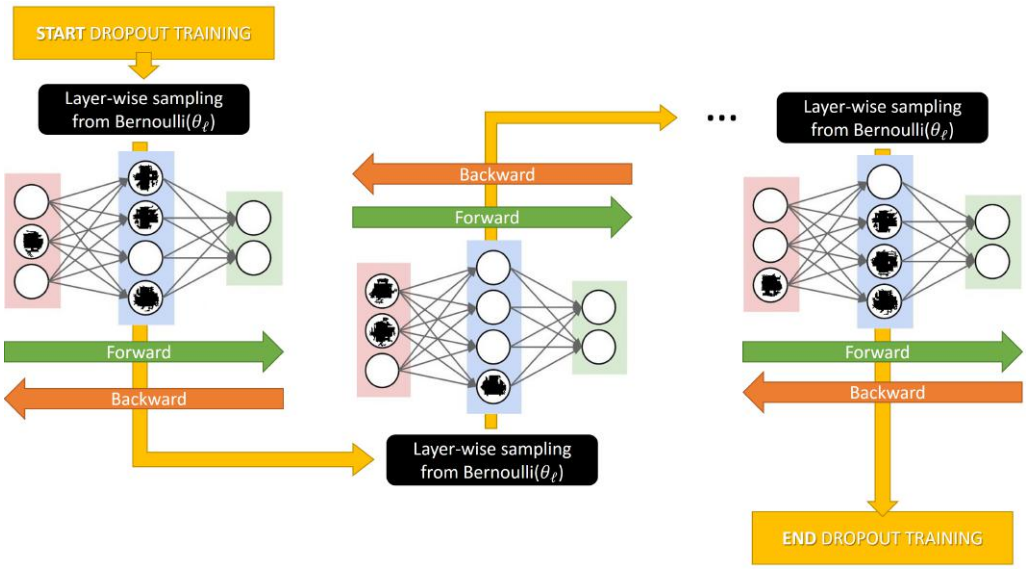


FIGURE 6.1: Dropout training for neural networks. It can be applied within the classical scheme of gradient descent one step of inference to compute network's predictions (forward pass) is applied to compute errors which are back-propagated in the following step of gradient updates (backward pass), being those two steps alternated until convergence. Dropout modifies the previous iterations by suppressing some units in the net - marked with a black spot - according to the realization of Bernoulli variables (if the sampled values equals 0, the unit is suppressed, being kept otherwise). Each layer has one hyperparameter θ_ℓ which is called *dropou rate* and which controls, on average, how many units are suppressed in that specific layer. Before each pairs of forward-backward passes, the Bernoulli variables are re-sampled and, since the network's weights are shared across all this subsamples, dropout can be interpreted as a model ensemble technique.

As a remedy for improve generalization capabilities of deep models, dropout has been proposed [Hin+12; Sri+14]. The idea is very simple and has been originally proposed for artificial neural networks which are nothing but weighted computational graphs where there exists a well established learning algorithm (called *backpropagation*) that implements gradient descent through the graph. Dropout can be embedded in this exact precise scheme. Precisely, it modifies each step of backpropagation (for weights' update) by a random deletion of units in the network according to the realization of a Bernoulli(θ) distribution, where θ is called *dropout rate* and can be changed from layer to layer. Thus, for each unit in a given layer, a Bernoulli-distributed random value is sampled and the network is erased from the network if and only if such value is 0. Then, within the remaining units, network's prediction are inferred (forward pass) and errors are back-propagated with one step of gradient descent. Afterwards, the units to be suppressed are changed (since the Bernoulli variables are re-sampled) and the previous scheme is repeated (see Figure 6.1).

Dropout has been proposed as an effective empirical technique to counteract overfitting since the random suppression scheme forces each unit of the network to "fire" more independently when input data are shown to the network.

This stage allows to simplify the inner representation computed by the network while, at the same time, circumventing the common problem of deep learning where networks are trained in settings where the number of trainable parameters is actually bigger than the amount of data that can be used for this task. Therefore, dropout has been designed as to counteract excessive correlations between units to occur since those correlations are postulated to be the exact cause of overfitting and overall degradation of generalization capabilities for the model. In other words, dropout prevents the net to memorize the dataset and, this is coherently effective in regularizing deep networks for which generalization problem need to be re-formulated with respect to classical shallow models [RethinkGeneralization].

Actually, despite the practical solidity of performance shown by dropout against overfitting, there is still a lack of understanding about what exact source of regularization it is promoted. In fact, despite a few recent works have been investigating such issue [Wag+13; HL15; BS13; BS14; Wag+14; GG16; Cav+17b]. However, for the sake of deploying their theoretical analysis, many of those applies simplification and approximations which, ultimately, may bias the overall understanding achieved. In this thesis, differently, we apply dropout training on the problem of matrix factorization which is the problem of approximating $\mathbf{X} \in \mathbb{R}^{m \times n}$, which is given, as $\mathbf{X} \approx \mathbf{U}\mathbf{V}^\top$. Here, we applied dropout on the factors $\mathbf{U} \in \mathbb{R}^{m \times d}$ and $\mathbf{V} \in \mathbb{R}^{n \times d}$ by randomly suppressing columns in both $\mathbf{U}$ and $\mathbf{V}$ according to the realizations of $\mathbf{r} \in \mathbb{R}^d$ whose entries are independently distributed as Bernoulli(θ), $0 < \theta < 1$. Within our analysis we are able to write down in analytical form the actual source of regularization which is induces by dropout and, when allowing the size of the factorization d is allowed to be variable, we discover connections with a classical low rank regularizer which, evidenty, allows us to fully grasp that dropout is achieving spectral sparsity - see Section 6.1.

Actually, the main technical tool that we employed during our analysis is the following. Since the dropout rate θ needs to be selected as an hyper-parameter, if we are able to adapt it to the data, we can actually avoid burdensome manual cross-validation while, possibly, improving the performance of the model. Inspired by this approach, we propose to modify the original dropout scheme for training deep neural networks. In fact, we conjecture that, due to the usual random initialization of the network's weight at the beginning of the training, overfitting it's unlike to appear during the first gradient updates. So, we should care about overfitting only eventually, when, at a certain point of the training, the net becomes overspecialized with respect to the training data and starts to memorize them. Since it's not easy to check when such problem starts to happen with a hard decision, we propose a soft scheduling of the dropout retain probability θ during training time t measured, say, in epochs. That is, we devise a simple implementation in the form of a time dependent function $\theta = \theta(t)$ so that, at the beginning of the training no unit is suppressed ($\theta(0) = 1$), when t increases $\theta(t)$ decreases in a smooth manner in order to gradually introduce dropout regularization within the model. We call this approach *Curriculum Dropout* [Mor+17] - see Section 6.2.

6.1 An analysis of Dropout for Matrix Factorization

In many problems in machine learning and artificial intelligence, no matter what the input dimensionality of the raw data is, relevant patterns and information often lie in a low-dimensional manifold. In order to capture its structure, linear subspaces have become very popular, arguably due to their efficiency and versatility [Lu+11].

Mathematically, a linear subspace is obtainable from data points $\mathbf{x}_1, \dots, \mathbf{x}_m \in \mathbb{R}^n$ as follows. We build the $m \times n$ matrix $\mathbf{X}$ that stacks each sample by rows. Then, when looking for a d-dimensional embedding, we search for two matrices, $\mathbf{U} \in \mathbb{R}^{m \times d}$ and $\mathbf{V} \in \mathbb{R}^{n \times d}$, such that $\mathbf{X} \approx \mathbf{U}\mathbf{V}^\top$. Algorithmically, $\mathbf{U}$ and $\mathbf{V}$ can be found through optimization, according to the *matrix factorization* (MF) problem

$$\min_{\mathbf{U},\mathbf{V}} \|\mathbf{X} - \mathbf{U}\mathbf{V}^\top\|_F^2 + \lambda\Omega(\mathbf{U}, \mathbf{V}) \tag{6.1}$$

where the Frobenius norm is a well established proxy to impose similarity between $\mathbf{X}$ and $\mathbf{U}\mathbf{V}^\top$. Also, for $\lambda > 0$, the regularizer $\Omega(\mathbf{U}, \mathbf{V})$ in (6.1) imposes some constraints on the factors: for instance, orthonormality as in PCA [Vid+16b].

Two are the main advantages of (6.1). First, we optimize on the factors directly, achieving a structured decomposition of $\mathbf{X}$. Second, the number of variables to be optimized scales linearly with respect to $m+n$, ensuring applicability even in the big data regime. Unfortunately, a big shortcoming in (6.1) arises. Indeed, when $\mathbf{U}$ is fixed, optimizing for $\mathbf{V}$ is a convex problem and vice versa, but, (6.1) is not convex when optimizing on $\mathbf{U}$ and $\mathbf{V}$ jointly. Therefore, one needs ancillary optimality conditions to ensure that the global optimum $(\mathbf{U}^{\text{opt}}, \mathbf{V}^{\text{opt}})$ of (6.1) exists as well as algorithms to compute a global optimum [Hae+14; HV15; HV17].

Those issues can be solved by replacing the MF problem (6.1) with *matrix approximation*, that is,

$$\min_{\mathbf{A}} \|\mathbf{X} - \mathbf{A}\|_F^2 + \gamma\Xi(\mathbf{A}). \tag{6.2}$$

In (6.2), $\gamma > 0$ and we minimize over $\mathbf{A} \in \mathbb{R}^{m \times n}$, forcing it to be close enough to $\mathbf{X}$ after adding the penalization term $\mathbf{X}$i which plays the analogous role on $\mathbf{A}$ as Ω does on $\mathbf{U}$ and $\mathbf{V}$ in (6.1).

The formulations in (6.1) and (6.2) are highly complementary. For instance, differently from (6.1), the optimization in (6.2) is convex and therefore, a global minimizer exists, is unique and can be found via gradient descent (and, sometimes, it has a closed-form solution, *e.g.* , when $\Xi = \|\cdot\|_F^2$). Again, differently from (6.1), the problem in (6.2) is not scalable (due to the $m \cdot n$ variables to be optimized) and, also, the optimal solution $\mathbf{A}^{\text{opt}}$ of (6.2) does not have the structure that (6.1) provides in terms of explicit factors $\mathbf{U}$ and $\mathbf{V}$.

In this Chapter, we bridge the gap between factorization (6.1) and approximation (6.2) for matrices, ultimately providing an unified framework by means of a recently developed strategy from deep learning: *dropout*.

Dropout [Hin+12; Sri+14] is a popular algorithm for training neural networks while preventing overfitting. During dropout training, each unit is endowed with a (binary) Bernoulli random variable of expected value θ - which is

called "retain probability". So, for each example/mini-batch, the network's weights are updated by using a back-propagation step which only involves the units whose corresponding Bernoulli variables are sampled with value 1. At each iteration, those Bernoulli variables are re-sampled again and the weights are updated accordingly. Note that, since all the sub-networks are sampled from the original architecture, the weights are shared across different units' subsamplings and dropout can be interpreted as a model ensemble. During inference, no units' suppression is performed and, simply, all the weights are rescaled by θ, the latter stage being interpreted as a model average up to certain approximations [Sri+14; BS13; BS14].

Motivated by the significant efforts made to understand dropout as (implicit) regularization [Wag+13; BS13; BS14; GG16], as in [ZZ15; Zhi+16], we combine dropout and MF through the following problem. While still looking for a direct optimization of $\mathbf{X} \approx \mathbf{U}\mathbf{V}^\top$ over factors $\mathbf{U} \in \mathbb{R}^{m\times d}$ and $\mathbf{V}^{n\times d}$, we replace (6.1) with

$$\min_{\mathbf{U},\mathbf{V}} \mathbb{E}_{\mathbf{r}} \left\| \mathbf{X} - \frac{1}{\theta}\mathbf{U}\mathrm{diag}(\mathbf{r})\mathbf{V}^\top \right\|_F^2 \tag{6.3}$$

where $\|\cdot\|_F$ denotes the Frobenius norm of a matrix $\mathbf{r} \in \mathbb{R}^d$ is a random vectors whose entries are i.i.d. Bernoulli(θ), and $\mathbb{E}_{\mathbf{r}}$ denotes the expected value with respect to $\mathbf{r}$. Essentially, by taking directly inspiration from the idea of suppressing "units" in a neural network, we here suppress "columns" of the factorization in order to obtain an optimization scheme that mimics the actual dropout training for neural networks. Indeed, in neural network training, batches of data are shaped as matrices and, when dropout is applied to the input layer, some columns of that matrix are set to zero. In practice, dropout for MF has shown solid performance [ZZ15; Zhi+16], but, it is still unclear what sort of regularization it induces for such class of problems.

The contributions of our theoretical analysis are the following:

1. We demonstrate that dropout for MF (6.3) is equivalent to the following deterministic regularization framework

$$\min_{\mathbf{U},\mathbf{V}} \left[\|\mathbf{X} - \mathbf{U}\mathbf{V}^\top\|_F^2 + \frac{1-\theta}{\theta}\Omega_{\text{dropout}}(\mathbf{U},\mathbf{V}) \right] \tag{6.4}$$

   where

$$\Omega_{\text{dropout}} = \sum_{k=1}^{d} \|\mathbf{u}_k\|_2^2 \|\mathbf{v}_k\|_2^2. \tag{6.5}$$

2. While carefully inspecting the nature of Ω_{dropout}, if we allow a variable size d of the factors $\mathbf{U}$ and $\mathbf{V}$, we observe that Ω_{dropout} naturally promotes for over-sized factorizations in the case of a fixed dropout rate θ.

3. We show that the regularizer induced by dropout acts as a low-rank regularization strategy. Specifically, we show that if the dropout rate θ is chosen as a given function of d, then the optimization problem in (6.3) is related to the following matrix approximation problem

$$\min_{\mathbf{A}} \left[\|\mathbf{X} - \mathbf{A}\|_F^2 + \gamma\|\mathbf{A}\|_\star^2 \right], \tag{6.6}$$

where the squared nuclear norm is used to induce low-rank factorizations.

4. Furthermore, if we are given the global optimum factors $\mathbf{U}^{\text{opt}}$ and $\mathbf{V}^{\text{opt}}$ of (6.3), then $\mathbf{A}^{\text{opt}} = (\mathbf{U}^{\text{opt}})(\mathbf{V}^{\text{opt}})^\top$ is the global optimum of (6.2) in the case of $\Xi(\mathbf{A}) = \|\mathbf{A}\|_\star^2$. Despite this result is derived in the case of variable size in the factorization, it is still applicable in the case of a fixed d.

6.1.1 Dropping out columns in matrix factorization

Given a fixed $m \times n$ matrix $\mathbf{X}$, we are interested in the problem of factorizing it as the product $\mathbf{U}\mathbf{V}^\top$, where $\mathbf{U}$ is $m \times d$ and $\mathbf{V}$ is $n \times d$, for some $d \geq \rho(\mathbf{X}) := \operatorname{rank}(\mathbf{X})$ that, in this Section, will be kept fixed for simplicity. In order to apply dropout to matrix factorization, we consider a random vector $\mathbf{r} = [r_1, \dots, r_d]$ whose elements are independently distributed as $r_i \sim \text{Bernoulli}(\theta)$.

Remark 1. *In what follows, to either avoid trivial cases or division by zero, we will assume* $0 < \theta < 1$. *Let us stress that, our perspective is more general than currently adopted practices for dropout training in neural networks where* $\theta > 0.5$ *(see [Sri+14, Appendix A.4] for a list of typical values).*

By means of $\mathbf{r}$, we can apply dropout to the problem $\min_{\mathbf{U},\mathbf{V}} \|\mathbf{X} - \mathbf{U}\mathbf{V}^\top\|_F^2$ as in (6.3). To see why the minimization of (6.3) can be achieved by dropping out columns of $\mathbf{U}$ and $\mathbf{V}$, observe that if we use a gradient descent strategy, the gradient of the expected value is equal to the expected value of the gradient. Therefore, if we choose a stochastic gradient descent (SGD) approach in which the expected gradient at each iteration is replaced by the gradient for a fixed sample $\mathbf{r}$, we obtain that, while moving from t-th to $(t+1)$-th iteration, the updated $\mathbf{U}^{(t+1)}, \mathbf{V}^{(t+1)}$ factors are computed accordingly to Algorithm 7. Thereby, the updates for the column of $\mathbf{U}^{(t+1)}, \mathbf{V}^{(t+1)}$ are either performed or skipped accordingly to $\mathbf{r}^{(t)}$. In fact, at t-th iteration, the columns of $\mathbf{U}$ and $\mathbf{V}$ for which $r_i^{(t)} = 0$ are not updated, and the gradient update is only applied to the columns for which $r_i^t = 1$. This observation precisely certifies that a SGD scheme[1] applied to (6.3) is actually implementing dropout as originally proposed in [Hin+12; Sri+14].

Corroborating the findings of various theoretical studies of dropout for general machine learning models [Hin+12; Sri+14; HL15; BS13; BS14; GG16; Hae+17a], we want to move tho the yet unexplored theory behind dropout for MF. Namely, we are interested in proving that the latter (6.3) is fully equivalent to a deterministic optimization problem of the form (6.1), for a particular choice of Ω. Ultimately, this will help us in better understanding of the implication of such random suppressions of columns that dropout is acting while the matrix $\mathbf{X}$ is factorized into $\mathbf{U}\mathbf{V}^\top$. This problem is tackled in the following theoretical result.

[1] Note that, when dropout training is applied in deep learning, the so-called *optimizer* (*e.g.* , ADAM [KB14]) needs to be fixed a priori and *independently* with respect to the usage of dropout. Therefore, our assumption of solving (6.3) with SGD is totally not-restrictive, being furthermore in line with the current implementation practices that are used for training deep neural networks (see [Tf]).

Algorithm 7: Dropout Training for MF

Randomly initialize $\mathbf{U}^{(0)}$ and $\mathbf{V}^{(0)}$ for a given $d > 0$. **foreach** $t = 1, 2, \dots$ **do**

Sample $\mathbf{r}^{(t)}$ elementwise from a Bernoulli(θ).

Compute the gradients

$$\begin{bmatrix} d\mathbf{U}^{(t)} \\ d\mathbf{V}^{(t)} \end{bmatrix} = \begin{bmatrix} (\mathbf{X} - \mathbf{V}^{(t)} \mathrm{diag}(\mathbf{r}^{(t)}) \mathbf{V}^{(t)\top}) \mathbf{V}^{(t)} \\ (\mathbf{X} - \mathbf{V}^{(t)} \mathrm{diag}(\mathbf{r}^{(t)}) \mathbf{V}^{(t)\top})^\top \mathbf{U}^{(t)} \end{bmatrix} \tag{6.7}$$

with respect to $\mathbf{U}$ and $\mathbf{V}$, respectively.

Update the factors

$$\begin{bmatrix} \mathbf{U}^{(t+1)} \\ \mathbf{V}^{(t+1)} \end{bmatrix} = \begin{bmatrix} \mathbf{U}^{(t)} \\ \mathbf{V}^{(t)} \end{bmatrix} + \frac{2\epsilon}{\theta} \begin{bmatrix} d\mathbf{U}^{(t)} \\ d\mathbf{V}^{(t)} \end{bmatrix} \mathrm{diag}(\mathbf{r}^{(t)}), \tag{6.8}$$

end

Theorem 9. *The two optimization problems (6.1) and (6.3) are equivalent while choosing λ and Ω in (6.3) to be*

$$\Omega_{dropout}(\mathbf{U}, \mathbf{V}) = \sum_{k=1}^{d} \|\mathbf{u}_k\|_2^2 \|\mathbf{v}_k\|_2^2, \tag{6.9}$$

where $\mathbf{u}_1, \dots, \mathbf{u}_d \in \mathbb{R}^m$ and $\mathbf{v}_1, \dots, \mathbf{v}_d \in \mathbb{R}^n$ stand for the columns of $\mathbf{U}$ and $\mathbf{V}$ respectively and $\lambda = \frac{1-\theta}{\theta}$.

Proof. In order to get the thesis, it is enough to prove that

$$\mathbb{E}_{\mathbf{r}} \|\theta \mathbf{X} - \mathbf{U} \mathrm{diag}(\mathbf{r}) \mathbf{V}^\top\|_F^2 = \theta^2 \|\mathbf{X} - \mathbf{U}\mathbf{V}^\top\|_F^2 + \theta(1-\theta) \sum_{k=1}^{d} \|\mathbf{u}_k\|_2^2 \|\mathbf{v}_k\|_2^2.$$

Since

$$\begin{aligned} &\mathbb{E}_{\mathbf{r}} \|\theta \mathbf{X} - \mathbf{U} \mathrm{diag}(\mathbf{r}) \mathbf{V}^\top\|_F^2 = \\ &= \mathbb{E}_{\mathbf{r}} \left\| \begin{bmatrix} \theta X_{11} - \sum_{k=1}^{d} U_{1k} r_k V_{1k}, & \dots, & \theta X_{1n} - \sum_{k=1}^{d} U_{1k} r_k V_{nk} \\ \vdots & \ddots & \vdots \\ \theta X_{m1} - \sum_{k=1}^{d} U_{mk} r_k V_{1k}, & \dots, & \theta X_{mn} - \sum_{k=1}^{d} U_{mk} r_k V_{nk} \end{bmatrix} \right\|_F^2, \end{aligned} \tag{6.10}$$

by definition of Frobenius norm and linearity of $\mathbb{E}_{\mathbf{r}}$, we elicit

$$\mathbb{E}_{\mathbf{r}} \|\theta \mathbf{X} - \mathbf{U} \mathrm{diag}(\mathbf{r}) \mathbf{V}^\top\|_F^2 = \sum_{i=1}^{m} \sum_{j=1}^{n} \mathbb{E}_{\mathbf{r}} \left[\left(\theta X_{ij} - \sum_{k=1}^{d} U_{ik} r_k V_{jk} \right)^2 \right]. \tag{6.11}$$

Use the bias-variance decomposition $\mathbb{E}[r^2] = \mathbb{V}[r] + \mathbb{E}[r]^2$, holding for a scalar random variable r.

$$\mathbb{E}_{\mathbf{r}}\|\theta\mathbf{X} - \mathbf{U}\mathrm{diag}(\mathbf{r})\mathbf{V}^\top\|_F^2 = \sum_{i=1}^{m}\sum_{j=1}^{n}\mathbb{V}_{\mathbf{r}}\left[\theta X_{ij} - \sum_{k=1}^{d} U_{ik}r_k V_{jk}\right] + \\ + \sum_{i=1}^{m}\sum_{j=1}^{n}\left(\mathbb{E}_{\mathbf{r}}\left[\theta X_{ij} - \sum_{k=1}^{d} U_{ik}r_k V_{jk}\right]\right)^2. \tag{6.12}$$

Since $r_1, \ldots, r_d$ are i.i.d., use the properties of expectation $\mathbb{E}_{\mathbf{r}}$ and variance $\mathbb{V}_{\mathbf{r}}$ with respect to linear combinations of independent random variables.

$$\mathbb{E}_{\mathbf{r}}\|\theta\mathbf{X} - \mathbf{U}\mathrm{diag}(\mathbf{r})\mathbf{V}^\top\|_F^2 = \sum_{i=1}^{m}\sum_{j=1}^{n}\sum_{k=1}^{d} U_{ik}^2 V_{jk}^2 \mathbb{V}_{\mathbf{r}}[r_k] + \\ + \sum_{i=1}^{m}\sum_{j=1}^{n}\left(\theta X_{ij} - \sum_{k=1}^{d} U_{ik}\mathbb{E}_{\mathbf{r}}[r_k] V_{jk}\right)^2. \tag{6.13}$$

Exploit the analytical formulas for expected value and variance of a Bernoulli(θ) distribution.

$$\mathbb{E}_{\mathbf{r}}\|\theta\mathbf{X} - \mathbf{U}\mathrm{diag}(\mathbf{r})\mathbf{V}^\top\|_F^2 = \sum_{i=1}^{m}\sum_{j=1}^{n}\sum_{k=1}^{d} U_{ik}^2 V_{jk}^2 \cdot \theta(1-\theta) + \\ + \sum_{i=1}^{m}\sum_{j=1}^{n}\left(\theta X_{ij} - \sum_{k=1}^{d} U_{ik}\cdot\theta\cdot V_{jk}\right)^2. \tag{6.14}$$

Rearrange the terms.

$$\mathbb{E}_{\mathbf{r}}\|\theta\mathbf{X} - \mathbf{U}\mathrm{diag}(\mathbf{r})\mathbf{V}^\top\|_F^2 = \theta(1-\theta)\sum_{k=1}^{d}\left(\sum_{i=1}^{m} U_{ik}^2\right)\left(\sum_{j=1}^{n} V_{jk}^2\right) + \\ + \theta^2\sum_{i=1}^{m}\sum_{j=1}^{n}\left(X_{ij} - \sum_{k=1}^{d} U_{ik}V_{jk}\right)^2. \tag{6.15}$$

Use the definition of row-by-column product of matrices

$$\mathbb{E}_{\mathbf{r}}\|\theta\mathbf{X} - \mathbf{U}\mathrm{diag}(\mathbf{r})\mathbf{V}^\top\|_F^2 = \theta(1-\theta)\sum_{k=1}^{d}\left(\sum_{i=1}^{m} U_{ik}^2\right)\left(\sum_{j=1}^{n} V_{jk}^2\right) + \\ + \theta^2\sum_{i=1}^{m}\sum_{j=1}^{n}\left(X_{ij} - \left[\mathbf{U}\mathbf{V}^\top\right]_{ij}\right)^2. \tag{6.16}$$

Apply the definitions of squared Euclidean norm $\|\cdot\|_2^2$ and Frobenius norm $\|\cdot\|_F$

$$\mathbb{E}_{\mathbf{r}}\|\theta\mathbf{X} - \mathbf{U}\mathrm{diag}(\mathbf{r})\mathbf{V}^\top\|_F^2 = \theta(1-\theta)\sum_{k=1}^{d}\|\mathbf{u}_k\|_2^2\|\mathbf{v}_k\|_2^2 + \theta^2\|\mathbf{X} - \mathbf{U}\mathbf{V}^\top\|_F.$$

This concludes the proof. □

Let us observe that, with the previous definition, λ can takes all possible non-negative scalar values, since as one can easily see, if we are interested in solving (6.1) with $\Omega = \Omega_{\text{dropout}}$ for a fixed λ value, we will always be able to find a fixed θ, $0 < \theta < 1$, such that $\lambda = \frac{1-\theta}{\theta}$. Indeed, since the relationship is invertible, one immediately gets $\theta = \frac{1}{1+\lambda}$.

The meaning of Theorem 9 is the following. Let consider the optimization problem (6.1) and fix $\Omega = \Omega_{\text{dropout}}$ as in (6.9) and $\lambda = \frac{1-\theta}{\theta}$. Then, the two optimization problems (6.1) and (6.3) are equivalent, where equivalence is intended in the strongest way possible, since for generic $\mathbf{U}$, $\mathbf{V}$, d and θ, we get

$$\begin{aligned}\mathbb{E}_{\mathbf{r}} \left\| \mathbf{X} - \frac{1}{\theta} \mathbf{U} \text{diag}(\mathbf{r}) \mathbf{V}^\top \right\|_F^2 = \\ = \|\mathbf{X} - \mathbf{U}\mathbf{V}^\top\|_F^2 + \frac{1-\theta}{\theta} \sum_{k=1}^{d} \|\mathbf{u}_k\|_2^2 \|\mathbf{v}_k\|_2^2. \end{aligned} \tag{6.17}$$

The implications of (6.17) are clear: any stationary point of (6.1) with $\Omega = \Omega_{\text{dropout}}$ is also a stationary point of (6.3) and vice versa. Furthermore, the two problems have the same global minimum since, despite the non convexity of the optimization problem, in the case of MF, there exist some theoretical guarantees to ensure the existence of a global minimizer due to the fact that the regularizer is shaped as product of columns of the factors [Rec+10; Hae+14; HV15; HV17]. For instance, while building on ideas derived from convex relaxations, general frameworks such as [HV15] allow for the analysis of non-convex factorizations and derives sufficient conditions for optimality condition of the non-convex optimization problem.

In this work, we characterize the optimum of droput with MF with a closed-form matrix approximation problem with squared nuclear norm regularization.

6.1.2 Connections with the nuclear norm

For $\mathbf{A} \in \mathbb{R}^{m \times n}$, its nuclear norm, also termed the trace norm or Schatten-Von Neumann 1-norm,

$$\|\mathbf{A}\|_\star = \sum_{i=1}^{\min(m,n)} \sigma_i(\mathbf{A}) \tag{6.18}$$

is defined as the sum of its singular values $\sigma_i(\mathbf{A})$, $i = 1, \ldots, \min(m, n)$. Within many machine learning problems [Yua+07; Arg+08; CR09; Cab+11; HO14], the usage of (6.18) is motivated by the fact that $\|\mathbf{A}\|_\star$ is a convex relaxation for the rank $\rho(\mathbf{A})$ of $\mathbf{A}$. Indeed, it is proved that the underlying low rank solution can be recovered by minimizing (6.18) under certain conditions [CT10; Rec+10].

In order to establish a connection between (6.18) and the regularizer (6.9), let us consider the following result.

Theorem 10 (*Variational form of the nuclear norm*).

$$\|\mathbf{X}\|_\star = \inf_{\mathbf{U},\mathbf{V}:\, \mathbf{U}\mathbf{V}^\top = \mathbf{X}} \sum_{k=1}^{d} \|\mathbf{u}_k\|_2 \|\mathbf{v}_k\|_2. \tag{6.19}$$

Proof. As a first stage, we have to prove that the Frobenius norm of a matrix is rotational invariant. That is, for any arbitrary $m \times n$ matrix $\mathbf{Y}$ and for any $m \times m$ orthonormal matrix $\mathbf{R}_1$ and $n \times n$ orthonormal matrix $\mathbf{R}_2$, we have

$$\|\mathbf{Y}\|_F = \|\mathbf{R}_1 \mathbf{Y} \mathbf{R}_2\|_F. \tag{6.20}$$

Fix $\mathbf{Y} \in \mathbb{R}^{m \times n}$, and two arbitrary matrices $\mathbf{R}_1$ and $\mathbf{R}_2$ as above. By observing that, for a rotational matrix, inverse and adjoint are equal, we get

$$\begin{aligned}
\|\mathbf{Y}\|_F^2 &= \langle \mathbf{Y}, \mathbf{Y} \rangle_F \\
&= \langle \mathbf{R}_1^\top \mathbf{R}_1 \mathbf{Y}, \mathbf{Y} \rangle_F \\
&= \langle \mathbf{R}_1^\top \mathbf{R}_1 \mathbf{Y}, \mathbf{Y} \mathbf{R}_2^\top \mathbf{R}_2 \rangle_F \\
&= \langle \mathbf{R}_1 \mathbf{Y} \mathbf{R}_2^\top, \mathbf{R}_1 \mathbf{Y} \mathbf{R}_2^\top \rangle_F \\
&= \|\mathbf{R}_1 \mathbf{Y} \mathbf{R}_2^\top\|_F^2.
\end{aligned}$$

Equation (6.20) follows by square-rooting the extremal members of the previous chain of inequalities.

As a necessary technical result, let us consider the family of Schatten/von Neumann ν-norms, $\nu \geq 1$, since the nuclear norm can be retrieved as a particular case. Fix $\nu \in [1, +\infty[$ arbitrary chosen[2]. For any $m \times n$ matrix $\mathbf{Y}$ we define the Schatten/von Neumann ν-norm

$$\|\mathbf{Y}\|_{\mathcal{S},\nu} = \left(\sum_{i=1}^{n} [\sigma_i(\mathbf{Y})]^\nu \right)^{1/\nu} \tag{6.21}$$

where $\sigma_i(\mathbf{Y})$ are the singular values of $\mathbf{Y}$ and we assume $m \geq n$, the latter hypothesis being non restrictive upon matrix inversion.
When $\nu = 2$, the Schatten/von Neumann norm $\|\cdot\|_{\mathcal{S},2}$ equals the Frobenius norm $\|\cdot\|_F$. Consider the singular value decomposition

$$\mathbf{Y} = \mathbf{L} \mathbf{\Sigma} \mathbf{R}^\top, \tag{6.22}$$

where Σ stacks $\sigma_1(\mathbf{Y}), \ldots, \sigma_n(\mathbf{Y})$ on the diagonal. By definition

$$\|\mathbf{Y}\|_{\mathcal{S},2}^2 = \sum_{i=1}^{n} [\sigma_i(\mathbf{Y})]^2 \tag{6.23}$$

and by definition of Frobenius norm

$$\|\mathbf{Y}\|_{\mathcal{S},2}^2 = \|\mathbf{\Sigma}\|_F^2. \tag{6.24}$$

By using (6.20) and (6.22),

$$\|\mathbf{\Sigma}\|_F^2 = \|\mathbf{L} \mathbf{\Sigma} \mathbf{R}^\top\|_F^2 = \|\mathbf{Y}\|_F^2, \tag{6.25}$$

[2] Although the case $\nu = +\infty$ is allowed in the literature, we will skip it here for simplicity

which is (6.21).

Now, in order to prove the variational form of the nuclear norm, we will prove it through a chain of equalities. First, we have that

$$\|\mathbf{Y}\|_\star = \min_{\mathbf{U},\mathbf{V}:\ \mathbf{U}\mathbf{V}^\top=\mathbf{Y}} \|\mathbf{U}\|_F\|\mathbf{V}\|_F \tag{6.26}$$

and we can prove it by showing that

$$\|\mathbf{Y}\|_\star \leq \min_{\mathbf{U},\mathbf{V}:\ \mathbf{U}\mathbf{V}^\top=\mathbf{Y}} \|\mathbf{U}\|_F\|\mathbf{V}\|_F \tag{6.27}$$

and

$$\min_{\mathbf{U},\mathbf{V}:\ \mathbf{U}\mathbf{V}^\top=\mathbf{Y}} \|\mathbf{U}\|_F\|\mathbf{V}\|_F \leq \|\mathbf{Y}\|_\star \tag{6.28}$$

hold at the same time. In order to prove (6.27), for a generic d, let $\mathbf{U}$ and $\mathbf{V}$ two $m \times d$ and $n \times d$ dimensional matrices, respectively, such that $\mathbf{Y} = \mathbf{U}\mathbf{V}^\top$. Then,

$$\|\mathbf{Y}\|_\star = \|\mathbf{U}\mathbf{V}^\top\|_\star \tag{6.29}$$

and

$$\|\mathbf{Y}\|_\star = \sum_{i=1}^{n} \sigma_i(\mathbf{U}\mathbf{V}^\top) \tag{6.30}$$

by definition of nuclear norm. Let

$$\rho = \operatorname{rank}(\mathbf{Y}) = \operatorname{rank}(\mathbf{U}\mathbf{V}^\top) \leq \min(\operatorname{rank}(\mathbf{U}), \operatorname{rank}(\mathbf{V})). \tag{6.31}$$

Then,

$$\|\mathbf{Y}\|_\star = \sum_{i=1}^{\rho} \sigma_i(\mathbf{U}\mathbf{V}^\top), \tag{6.32}$$

since $\sigma_{\rho+1}(\mathbf{Y}) = \cdots = \sigma_n(\mathbf{Y}) = 0$. Use Von Neumann inequality

$$\|\mathbf{Y}\|_\star \leq \sum_{i=1}^{\rho} \sigma_i(\mathbf{U})\sigma_i(\mathbf{V}) \tag{6.33}$$

Apply Cauchy-Schwartz inequality.

$$\|\mathbf{Y}\|_\star \leq \sqrt{\sum_{i=1}^{\rho} \sigma_i(\mathbf{U})^2}\sqrt{\sum_{j=1}^{\rho} \sigma_j(\mathbf{V})^2}. \tag{6.34}$$

The square-rooting is an increasing function. Therefore, through (6.31),

$$\|\mathbf{Y}\|_\star \leq \sqrt{\sum_{i=1}^{\rho} \sigma_i(\mathbf{U})^2}\sqrt{\sum_{j=1}^{\operatorname{rank}(\mathbf{V})} \sigma_j(\mathbf{V})^2} \leq \sqrt{\sum_{i=1}^{\operatorname{rank}(\mathbf{U})} \sigma_i(\mathbf{U})^2}\sqrt{\sum_{j=1}^{\operatorname{rank}(\mathbf{V})} \sigma_j(\mathbf{V})^2}, \tag{6.35}$$

since adding non-negative addends to the summations. Hence, using (6.21) with $\nu = 2$,

$$\|\mathbf{Y}\|_\star \leq \|\mathbf{U}\|_{\mathcal{S},2}\|\mathbf{V}\|_{\mathcal{S},2}, \tag{6.36}$$

but now

$$\|\mathbf{Y}\|_\star \leq \|\mathbf{U}\|_F\|\mathbf{V}\|_F. \tag{6.37}$$

and

$$\|\mathbf{Y}\|_\star \leq \|\mathbf{U}\|_F \|\mathbf{V}\|_F. \tag{6.38}$$

Since d, $\mathbf{U}$ and $\mathbf{V}$ are generic, we can minimize both terms with respect to them. In this case, the inequality is trivially preserved. Then,

$$\|\mathbf{Y}\|_\star \leq \min_{\mathbf{U},\mathbf{V}:\, \mathbf{U}\mathbf{V}^\top = \mathbf{Y}} \|\mathbf{U}\|_F \|\mathbf{V}\|_F, \tag{6.39}$$

obtaining (6.27). Similarly, in order to prove (6.28), let us consider *a* thin singular value decomposition $\mathbf{Y} = \mathbf{L}\mathbf{\Sigma}\mathbf{R}^\top$ for $\mathbf{Y}$ and let us choose

$$\overline{\mathbf{U}} = \mathbf{L}\mathbf{\Sigma}^{1/2} \quad \text{and} \quad \overline{\mathbf{V}} = \mathbf{R}\mathbf{\Sigma}^{1/2}, \tag{6.40}$$

being $\mathbf{\Sigma}^{1/2}$ the diagonal matrix obtaining from $\mathbf{\Sigma}$ entrywise square-rooting all its entries. Note that

$$\overline{\mathbf{U}}\,\overline{\mathbf{V}}^\top = \mathbf{L}\mathbf{\Sigma}^{1/2}\left(\mathbf{R}\mathbf{\Sigma}^{1/2}\right)^\top = \mathbf{L}\mathbf{\Sigma}^{1/2}\mathbf{\Sigma}^{1/2}\mathbf{R}^\top = \mathbf{L}\mathbf{\Sigma}\mathbf{R}^\top = \mathbf{Y}. \tag{6.41}$$

Hence,

$$\min_{\mathbf{U},\mathbf{V}:\, \mathbf{U}\mathbf{V}^\top = \mathbf{Y}} \|\mathbf{U}\|_F \|\mathbf{V}\|_F \leq \|\overline{\mathbf{U}}\|_F \|\overline{\mathbf{V}}\|_F \tag{6.42}$$

and

$$\min_{\mathbf{U},\mathbf{V}} \|\mathbf{U}\|_F \|\mathbf{V}\|_F \leq \|\mathbf{\Sigma}^{1/2}\|_F \|\mathbf{\Sigma}^{1/2}\|_F \tag{6.43}$$

by using (6.20) in the right member. Then

$$\min_{\mathbf{U},\mathbf{V}:\, \mathbf{U}\mathbf{V}^\top = \mathbf{Y}} \|\mathbf{U}\|_F \|\mathbf{V}\|_F \leq \left(\|\mathbf{\Sigma}^{1/2}\|_F\right)^2. \tag{6.44}$$

Observe that

$$\left(\|\mathbf{\Sigma}^{1/2}\|_F\right)^2 = \sum_{i=1}^{\operatorname{rank}(\mathbf{Y})} \left(\sqrt{\sigma_i(\mathbf{Y})}\right)^2 = \sum_{i=1}^{\operatorname{rank}(\mathbf{Y})} \sigma_i(\mathbf{Y}) = \operatorname{trace}(\mathbf{\Sigma}) \tag{6.45}$$

Thus, by means of the definition of the nuclear norm,

$$\min_{\mathbf{U},\mathbf{V}:\, \mathbf{U}\mathbf{V}^\top = \mathbf{Y}} \|\mathbf{U}\|_F \|\mathbf{V}\|_F \leq \operatorname{trace}(\mathbf{\Sigma}) = \|\mathbf{Y}\|_\star \tag{6.46}$$

which gives (6.28).

We can achieve one other equivalent expression for the nuclear norm. In order to do this, let us consider the following necessary technical result. For any $a, b \in \mathbb{R}$

$$2ab = \min_{\eta > 0} \left(\frac{a^2}{\eta} + \eta b^2\right) \tag{6.47}$$

In fact, Fix arbitrary $a, b \in \mathbb{R}$ and $\eta > 0$. Then

$$0 \leq (a - \eta b)^2,$$

$$0 \leq a^2 - 2\eta ab + \eta^2 b^2,$$

$$2\eta ab \leq a^2 + \eta^2 b^2.$$

Divide each term by $\eta > 0$.

$$2ab \leq \frac{a^2}{\eta} + \eta b^2.$$

The inequality is preserved after p-th order exponentiation, $p > 0$.

$$(2ab) \leq \left(\frac{a^2}{\eta} + \eta b^2\right).$$

Thus, the function $f_{a,b}(\eta) = (\frac{a^2}{\eta} + \eta b^2)$ upper bounds $(2ab)$ for any a, b, η. Then, since

$$f_{a,b}(a/b) = \left(\frac{a^2}{a}b + \frac{a}{b}b^2\right) = (ab + ab) = (2ab), \tag{6.48}$$

we conclude.

Therefore, we obtain

$$\|\mathbf{Y}\|_\star = \min_{\mathbf{U},\mathbf{V}:\, \mathbf{U}\mathbf{V}^\top = \mathbf{Y}} \frac{1}{2}\left(\|\mathbf{U}\|_F^2 + \|\mathbf{V}\|_F^2\right) \tag{6.49}$$

It is enough to observe that, for any matrix $\mathbf{U}$ and $\mathbf{V}$ such that $\mathbf{U}\mathbf{V}^\top = \mathbf{Y}$, also $\mathbf{U}/\sqrt{\eta}$ and $\sqrt{\eta}\mathbf{V}$ still satisfy the same property, $\eta > 0$. Then, for any arbitrary $\eta > 0$,

$$\min_{\mathbf{U},\mathbf{V}:\, \mathbf{U}\mathbf{V}^\top = \mathbf{Y}} \frac{1}{2}\left(\|\mathbf{U}\|_F^2 + \|\mathbf{V}\|_F^2\right) = \min_{\mathbf{U},\mathbf{V}:\, \mathbf{U}\mathbf{V}^\top = \mathbf{Y}} \left(\frac{1}{2}\frac{\|\mathbf{U}\|_F^2}{\eta} + \frac{1}{2}\eta\|\mathbf{V}\|_F^2\right) \tag{6.50}$$

Minimize over η and apply (6.47).

$$\min_{\mathbf{U},\mathbf{V}:\, \mathbf{U}\mathbf{V}^\top = \mathbf{Y}} \frac{1}{2}\left(\|\mathbf{U}\|_F^2 + \|\mathbf{V}\|_F^2\right) = \min_{\mathbf{U},\mathbf{V}:\, \mathbf{U}\mathbf{V}^\top = \mathbf{Y}} \left(\|\mathbf{U}\|_F \|\mathbf{V}\|_F\right). \tag{6.51}$$

At the same time,

$$\|\mathbf{Y}\|_\star = \min_{\mathbf{U},\mathbf{V}:\, \mathbf{U}\mathbf{V}^\top = \mathbf{Y}} \frac{1}{2}\left(\sum_{k=1}^{d} \|\mathbf{u}_k\|_2^2 + \sum_{k=1}^{d} \|\mathbf{v}_k\|_2^2\right), \tag{6.52}$$

where $\mathbf{u}_k \in \mathbb{R}^m$ and $\mathbf{v}_k \in \mathbb{R}^n$ denote the k-th column of $\mathbf{U}$ and $\mathbf{V}$, respectively. In fact (6.52) follows by applying (6.49) and observing that

$$\|\mathbf{U}\|_F^2 + \|\mathbf{V}\|_F^2 = \sum_{k=1}^{d} \|\mathbf{u}_k\|_2^2 + \sum_{k=1}^{d} \|\mathbf{v}_k\|_2^2. \tag{6.53}$$

On top of that,

$$\|\mathbf{Y}\|_\star = \min_{\mathbf{U},\mathbf{V}:\, \mathbf{U}\mathbf{V}^\top = \mathbf{Y}} \sum_{k=1}^{d} \|\mathbf{u}_k\|_2 \|\mathbf{v}_k\|_2, \tag{6.54}$$

where $\mathbf{u}_k \in \mathbb{R}^m$ and $\mathbf{v}_k \in \mathbb{R}^n$ denote the k-th column of $\mathbf{U}$ and $\mathbf{V}$, respectively.

Hence, Apply (6.47) for any $k = 1, \dots, d$. Then,

$$\|\mathbf{u}_k\|_2 \|\mathbf{v}_k\|_2 = \min_{\eta_k > 0} \left(\frac{1}{2} \frac{\|\mathbf{u}_k\|_2^2}{\eta_k} + \eta_k \|\mathbf{v}_k\|_2^2 \right). \tag{6.55}$$

Since we have a decoupled minimization problem, we can commute summation and minimum, rewriting

$$\sum_{k=1}^{d} \|\mathbf{u}_k\|_2 \|\mathbf{v}_k\|_2 = \sum_{k=1}^{d} \min_{\eta_k > 0} \left(\frac{1}{2} \frac{\|\mathbf{u}_k\|_2^2}{\eta_k} + \eta_k \|\mathbf{v}_k\|_2^2 \right) \tag{6.56}$$

into

$$\sum_{k=1}^{d} \|\mathbf{u}_k\|_2 \|\mathbf{v}_k\|_2 = \min_{\eta_1, \dots, \eta_d > 0} \left(\frac{1}{2} \sum_{k=1}^{d} \frac{\|\mathbf{u}_k\|_2^2}{\eta_k} + \sum_{k=1}^{d} \eta_k \|\mathbf{v}_k\|_2^2 \right). \tag{6.57}$$

Minimize both terms with respect to d, $\mathbf{U}$ and $\mathbf{V}$.

$$\begin{aligned} &\min_{\mathbf{U},\mathbf{V}\colon \mathbf{U}\mathbf{V}^\top = \mathbf{Y}} \sum_{k=1}^{d} \|\mathbf{u}_k\|_2 \|\mathbf{v}_k\|_2 \\ &= \min_{\eta_1, \dots, \eta_d > 0, d, \mathbf{U}, \mathbf{V}\colon \mathbf{U}\mathbf{V}^\top = \mathbf{Y}} \left(\frac{1}{2} \sum_{k=1}^{d} \frac{\|\mathbf{u}_k\|_2^2}{\eta} + \eta \|\mathbf{v}_k\|_2^2 \right). \end{aligned} \tag{6.58}$$

Equivalently,

$$\begin{aligned} &\min_{\mathbf{U},\mathbf{V}\colon \mathbf{U}\mathbf{V}^\top = \mathbf{Y}} \sum_{k=1}^{d} \|\mathbf{u}_k\|_2 \|\mathbf{v}_k\|_2 \\ &= \min_{\eta_1, \dots, \eta_d > 0, d, \mathbf{U}, \mathbf{V}\colon \mathbf{U}\mathbf{V}^\top = \mathbf{Y}} \left(\frac{1}{2} \sum_{k=1}^{d} \|\mathbf{u}_k / \sqrt{\eta_k}\|_2^2 + \|\sqrt{\eta_k} \mathbf{v}_k\|_2^2 \right). \end{aligned} \tag{6.59}$$

We can discard the factors η_k exploiting the condition $\mathbf{U}\mathbf{V}^\top = \mathbf{Y}$.

$$\min_{\mathbf{U},\mathbf{V}\colon \mathbf{U}\mathbf{V}^\top = \mathbf{Y}} \sum_{k=1}^{d} \|\mathbf{u}_k\|_2 \|\mathbf{v}_k\|_2 = \min_{d, \mathbf{U}, \mathbf{V}\colon \mathbf{U}\mathbf{V}^\top = \mathbf{Y}} \left(\frac{1}{2} \sum_{k=1}^{d} \|\mathbf{u}_k\|_2^2 + \|\mathbf{v}_k\|_2^2 \right).$$

We get

$$\min_{\mathbf{U},\mathbf{V}\colon \mathbf{U}\mathbf{V}^\top = \mathbf{Y}} \sum_{k=1}^{d} \|\mathbf{u}_k\|_2 \|\mathbf{v}_k\|_2 = \|\mathbf{Y}\|_\star$$

by means of (6.52).

Due to the fact that, in all previous formulas, d was fixed and generic, the variational forms still apply if we select d to be the arg minimum of the objective functions whose minimization over $\mathbf{U}$ and $\mathbf{V}$ gives the variational

Algorithm 8: Pathological oversizing in the factors

Randomly initialize $\mathbf{U}^{(0)}$ and $\mathbf{V}^{(0)}$ for a given $d > 0$.
foreach $t = 1, 2, \dots$ **do**
  Perform the following update for the factors

$$\begin{aligned} \mathbf{U}^{(t+1)} &= \frac{\sqrt{2}}{2}\left[\mathbf{U}^{(t)}, \mathbf{U}^{(t)}\right] \\ \mathbf{V}^{(t+1)} &= \frac{\sqrt{2}}{2}\left[\mathbf{V}^{(t)}, \mathbf{V}^{(t)}\right] \end{aligned} \tag{6.66}$$

end

forms itself. Therefore,

$$\|\mathbf{X}\|_\star = \min_{d \geq \rho(\mathbf{X}), \mathbf{U}\in\mathbb{R}^{m\times d}, \mathbf{V}\in\mathbb{R}^{n\times d}:\, \mathbf{U}\mathbf{V}^\top = \mathbf{X}} \frac{1}{2}\left(\|\mathbf{U}\|_F^2 + \|\mathbf{V}\|_F^2\right) \tag{6.60}$$

$$= \min_{d \geq \rho(\mathbf{X}), \mathbf{U}\in\mathbb{R}^{m\times d}, \mathbf{V}\in\mathbb{R}^{n\times d}:\, \mathbf{U}\mathbf{V}^\top = \mathbf{X}} \|\mathbf{U}\|_F \|\mathbf{V}\|_F \tag{6.61}$$

$$= \min_{d \geq \rho(\mathbf{X}), \mathbf{U}\in\mathbb{R}^{m\times d}, \mathbf{V}\in\mathbb{R}^{n\times d}:\, \mathbf{U}\mathbf{V}^\top = \mathbf{X}} \frac{1}{2}\left(\sum_{k=1}^{d} \|\mathbf{u}_k\|_2^2 + \sum_{k=1}^{d} \|\mathbf{v}_k\|_2^2\right) \tag{6.62}$$

$$= \min_{d \geq \rho(\mathbf{X}), \mathbf{U}\in\mathbb{R}^{m\times d}, \mathbf{V}\in\mathbb{R}^{n\times d}:\, \mathbf{U}\mathbf{V}^\top = \mathbf{X}} \sum_{k=1}^{d} \|\mathbf{u}_k\|_2 \|\mathbf{v}_k\|_2. \tag{6.63}$$

This concludes the proof of Theorem 10 □

We can find a close similarity between computing the infimum of (6.9) over $\mathbf{U}$ and $\mathbf{V}$ such that $\mathbf{U}\mathbf{V}^\top = \mathbf{X}$ and (6.19), except to a point. Instead of summing the product Euclidean norms $\|\cdot\|_2$ among the columns of $\mathbf{U}$ and $\mathbf{V}$ as in (6.19), in Ω_{dropout}, we are summing the products of *squared* Euclidean norms $\|\cdot\|_2^2$ among the columns of $\mathbf{U}$ and $\mathbf{V}$. Although this difference may seem marginal, this is not actually the case.

Remark 2. *Let fix two arbitrary random matrices* $\mathbf{U}$ *and* $\mathbf{V}$ *of sizes* $m \times d$ *and* $n \times d$, *respectively. Now, consider the case of a variable size of factorization* d. *Then,*

$$0 = \inf_{d,\mathbf{U},\mathbf{V}} \sum_{k=1}^{d} \|\mathbf{u}_k\|_2^2 \|\mathbf{v}_k\|_2^2 \text{qquads.t.} \quad \begin{cases} d \geq \rho(\mathbf{X}) \\ \mathbf{U}\mathbf{V}^\top = \mathbf{X} \end{cases} \tag{6.64}$$

since we can observe that

$$\Omega_{dropout}\left(\frac{\sqrt{2}}{2}[\mathbf{U},\mathbf{U}], \frac{\sqrt{2}}{2}[\mathbf{V},\mathbf{V}]\right) = \frac{1}{2}\Omega_{dropout}\left(\mathbf{U},\mathbf{V}\right). \tag{6.65}$$

So if we minimize the objective function (6.3) - or, equivalently, (6.1) with $\Omega = \Omega_{dropout}$ *- over* $\mathbf{U}$,$\mathbf{V}$ *and* d *as well, we may trivially lower the value of the objective function through Algorithm 8 which, clearly does not promote* $\mathbf{U}\mathbf{V}^\top$ *to be close to* $\mathbf{X}$ *in any case.*

Proof. Let $\mathbf{U}$ and $\mathbf{V}$ such that $\mathbf{U}\mathbf{V}^\top = \mathbf{X}$ for a particular choice of d. Denote

$$\Omega(\mathbf{U}, \mathbf{V}) = \sum_{k=1}^{d} \|\mathbf{u}_k\|_2^2 \|\mathbf{v}_k\|_2^2 \tag{6.67}$$

and define

$$\mathbf{A} = \frac{\sqrt{2}}{2}[\mathbf{U}, \mathbf{U}] \in \mathbb{R}^{m \times 2d} \tag{6.68}$$

$$\mathbf{B} = \frac{\sqrt{2}}{2}[\mathbf{V}, \mathbf{V}] \in \mathbb{R}^{n \times 2d}. \tag{6.69}$$

Then

$$\mathbf{A}\mathbf{B}^\top = \left(\frac{\sqrt{2}}{2}\right)^2 \mathbf{U}\mathbf{V}^\top + \left(\frac{\sqrt{2}}{2}\right)^2 \mathbf{U}\mathbf{V}^\top = \frac{1}{2}\mathbf{X} + \frac{1}{2}\mathbf{X} = \mathbf{X} \tag{6.70}$$

and

$$\Omega(\mathbf{A}, \mathbf{B}) = \sum_{k=1}^{2d} \|\mathbf{a}_k\|_2^2 \|\mathbf{b}_k\|_2^2 \tag{6.71}$$

$$= \frac{1}{4}\sum_{k=1}^{d} \|\mathbf{u}_k\|_2^2 \|\mathbf{v}_k\|_2^2 + \frac{1}{4}\sum_{k=1}^{d} \|\mathbf{u}_k\|_2^2 \|\mathbf{v}_k\|_2^2 = \frac{1}{2}\Omega(\mathbf{U}, \mathbf{V}). \tag{6.72}$$

In light of this observation, suppose, by absurd that $\varepsilon > 0$ is the minimum of (4.23), being such value realizes for some matrix $\mathbf{U}$ and $\mathbf{V}$. Then, we can repeat the same construction and produce a pairs of matrix $\mathbf{A}$ and $\mathbf{B}$ such that $\Omega(\mathbf{A}, \mathbf{B}) = \frac{\varepsilon}{2}$. Thus, necessarily, (4.23) holds being the objective non-negative. □

In the previous observation, we analyzed what happens if we relax d from a fixed and (heuristically) chosen value to be one of the active variables of the optimization. The latter aspect is actively investigates as a research topic [Rec+10; Hae+14; HV15; HV17; Hae+17a] and many algorithms have been proposed with this respect so that, in our case, we can take advantage of any of those when asked to optimize

$$\min_{\mathbf{U},\mathbf{V},d} \mathbb{E}_{\mathbf{r}} \left\| \mathbf{X} - \frac{1}{\theta}\mathbf{U}\mathrm{diag}(\mathbf{r})\mathbf{V}^\top \right\|_F^2 \tag{3'}$$

over d as well. In the present work, we will not investigate this aspect, since it's not primarily related to our scope. Differently, we allow d to be variable for the sake of improving the theoretical understanding dropout of MF. Through this modification, additionally, we bridge the gap between dropout for MF (6.3), its equivalent reformulation (6.1) with $\Omega = \Omega_{\text{dropout}}$ and the matrix factorization problem (6.2) where $\Xi(\mathbf{A}) = \|\mathbf{A}\|_\star^2$.

6.1.3 Variable size factors

In this Section, we want to establish a connection between the class of problems (6.2) and dropout for MF, as explained in the previous Section can be

formulated either as (6.3) or as its fully deterministic counterpart (6.1) with Ω as in (6.9).

In order to fill such gap, we are interested in observing whether there exists a way to choose θ to depend upon the size of the factorization d, such that we can avoid the pathological optimization scheme of Algorithm 8 which promotes over-sized factorizations.

Proposition 4. *For a given* p, $0 < p < 1$, *define*

$$\theta(d) = \frac{p}{d - (d-1)p} \tag{6.73}$$

where d *refers to the size of the factorization for* $\mathbf{X}$, *quantified in terms of columns of* $\mathbf{U}$ *and* $\mathbf{V}$. *Then*

$$\frac{1-\theta(2d)}{\theta(2d)}\Omega_{dropout}\left(\frac{\sqrt{2}}{2}[\mathbf{U},\mathbf{U}], \frac{\sqrt{2}}{2}[\mathbf{V},\mathbf{V}]\right) = \frac{1-\theta(d)}{\theta(d)}\Omega_{dropout}(\mathbf{U},\mathbf{V}).$$

Proof. We will prove $\theta(d) > 0$ and $\theta(d) < 1$ separately. Since $p > 0$, then $\theta(d) > 0$ if and only if $m - (m-1)p > 0$. But this is true since

$$m - (m-1)p = m - mp + p \geq m(1-p) > 0. \tag{6.74}$$

On the other hand, since the fraction $\theta(d)$ is positive, $\theta(d) < 1$ is verified if and only if

$$p < m - (m-1)p \tag{6.75}$$

if and only if

$$0 < m - mp \tag{6.76}$$

if and only if

$$p < 1 \tag{6.77}$$

which is actually true by assumption. The property can also be verified analytically by noticing that

$$\frac{1-\theta(d)}{\theta(d)} = \frac{1 - \frac{p}{d-(d-1)p}}{\frac{p}{d-(d-1)p}} = \frac{d-(d-1)p-p}{p} = d\frac{1-p}{p}. \tag{6.78}$$

□

In Proposition 4, we modify the dropout retain probability θ to be function of d, while also depending on a novel hyper-parameter p. We will discuss later on the meaning and the necessity of introducing it, but for now, let's say that p is fixed in the range $]0,1[$.

In principle, the only guarantee that Proposition 4 ensures is that the choice $\theta = \theta(d)$ as in 6.73 prevents the over-sizing in the factorization. Indeed, other issues may arise and, potentially, one may be asked to change $\theta(d)$ in order to accommodate for them. Actually, we can show that the definition (6.73) is able to solve *all* the problematics of dropout applied to MF with variable size due to the following result.

Proposition 5. *For* $\theta = \theta(d)$ *as defined in* (6.73), $\frac{1-p}{p}\|\mathbf{X}\|_\star^2$ *is the lower convex envelope*[3] *of*

$$\Lambda(\mathbf{X}) = \inf_{U,V,d}\left[\frac{1-\theta(d)}{\theta(d)}\Omega_{dropout}(\mathbf{U},\mathbf{V})\right]$$

subject to $d \geq \rho(\mathbf{X})$ *and* $\mathbf{U}\mathbf{V}^\top = \mathbf{X}$.

Proof. First, recall that the convex envelope of a function f is the largest closed, convex function g such that $g(x) \leq f(x)$ for all x and is given by $g = (f^*)^*$, where f^* denotes the Fenchel dual of f, defined as $f^*(q) \equiv \sup_x \langle q, x\rangle - f(x)$. Let $\Theta(\mathbf{X}) = \frac{1}{2}\|\mathbf{X}\|_\triangle^2$, given by

$$\Theta(\mathbf{X}) = \inf_{\substack{d \geq \rho(\mathbf{X}) \\ \mathbf{U}\in\mathbb{R}^{m\times d} \\ \mathbf{V}\in\mathbb{R}^{n\times d} \\ \text{s.t. } \mathbf{U}\mathbf{V}^\top = \mathbf{X}}} \frac{\lambda_d}{2}\sum_{k=1}^{d}\|\mathbf{u}_k\|_2^2\|\mathbf{v}_k\|_2^2. \tag{6.79}$$

and note that this can be equivalently written by the equation

$$\Theta(\mathbf{X}) = \inf_{\substack{d \geq \rho(\mathbf{X}) \\ \mathbf{U}\in\mathbb{R}^{m\times d} \\ \mathbf{V}\in\mathbb{R}^{n\times d} \\ \Lambda\in\mathbb{R}^d}} \frac{\lambda_d}{2}\|\Lambda\|_2^2 \;\text{ s.t. }\; \sum_{k=1}^{d}\Lambda_k\mathbf{u}_k\mathbf{v}_k^\top = \mathbf{X} \;\text{ and }\; (\|\mathbf{u}_k\|_2, \|\mathbf{v}_k\|_2) \leq (1,1)\;\; \forall k. \tag{6.80}$$

This gives the Fenchel dual of Θ as

$$\Theta^*(\mathbf{Q}) = \sup_d \sup_{\substack{\mathbf{U}\in\mathbb{R}^{m\times d} \\ \mathbf{V}\in\mathbb{R}^{n\times d} \\ \Lambda\in\mathbb{R}^d}} \sum_{k=1}^{d}\Lambda_k\left\langle \mathbf{Q}, \mathbf{u}_k\mathbf{v}_k^\top\right\rangle - \frac{\lambda_d}{2}\|\Lambda\|_2^2 \;\text{ s.t. }\; (\|\mathbf{u}_k\|_2, \|\mathbf{v}_k\|_2) \leq (1,1)\;\; \forall k. \tag{6.81}$$

Now, note that if we define the vector $\mathbf{B}_d(\mathbf{U},\mathbf{V}) \in \mathbb{R}^d$ as

$$\mathbf{B}_d(\mathbf{U},\mathbf{V}) = \begin{bmatrix} \left\langle \mathbf{Q}, \mathbf{u}_1\mathbf{v}_1^\top\right\rangle \\ \left\langle \mathbf{Q}, \mathbf{u}_2\mathbf{v}_2^\top\right\rangle \\ \vdots \\ \left\langle \mathbf{Q}, \mathbf{u}_d\mathbf{v}_d^\top\right\rangle \end{bmatrix}, \tag{6.82}$$

[3] One defines lower convex envelope of a function f as the supremum over all convex functions g such that $g \leq f$.

then from (6.81) we have that, for every k,

$$\Theta^*(\mathbf{Q}) = \sup_{d} \sup_{\substack{\mathbf{U}\in\mathbb{R}^{m\times d}\\ \mathbf{V}\in\mathbb{R}^{n\times d}}} \sup_{\Lambda\in\mathbb{R}^d} \langle \mathbf{B}_d(\mathbf{U},\mathbf{V}),\Lambda\rangle - \frac{\lambda_d}{2}\|\Lambda\|_2^2 \quad \text{s.t.} \quad (\|\mathbf{u}_k\|_2, \|\mathbf{v}_k\|_2) \leq (1,1) \tag{6.83}$$

$$= \sup_{d} \sup_{\substack{\mathbf{U}\in\mathbb{R}^{m\times d}\\ \mathbf{V}\in\mathbb{R}^{n\times d}}} \frac{1}{2\lambda_d}\|\mathbf{B}_d(\mathbf{U},\mathbf{V})\|_2^2 \quad \text{s.t.} \quad (\|\mathbf{u}_k\|_2, \|\mathbf{v}_k\|_2) \leq (1,1). \tag{6.84}$$

where the final equality comes from noting that the supremum w.r.t. Λ is the definition of the Fenchel dual of the squared ℓ_2 norm evaluated at $\mathbf{B}_d(\mathbf{U},\mathbf{V})$.

Now, from(6.84) and the definition of $\mathbf{B}_d(\mathbf{U},\mathbf{V})$ note that for a fixed value of d, (6.84) is optimized w.r.t. $(\mathbf{U},\mathbf{V})$ by choosing all the columns of $(\mathbf{U},\mathbf{V})$ to be equal to the maximum singular vector pair, given by

$$\sup_{\mathbf{u}\in\mathbb{R}^m,\mathbf{v}\in\mathbb{R}^n} \left\langle \mathbf{Q}, \mathbf{u}\mathbf{v}^\top \right\rangle \quad \text{s.t.} \quad (\|\mathbf{u}\|_2, \|\mathbf{v}\|_2) \leq (1,1). \tag{6.85}$$

Note also that for this optimal choice of $(\mathbf{U},\mathbf{V})$ we have that $\mathbf{B}_d(\mathbf{U},\mathbf{V}) = \sigma(\mathbf{Q})\mathbf{1}_d$ where $\sigma(\mathbf{Q})$ denotes the largest singular value of $\mathbf{Q}$ and $\mathbf{1}_d$ is a vector of all ones of size d. Plugging this in (6.84) gives

$$\Theta^*(\mathbf{Q}) = \sup_d \frac{1}{2\lambda_d}\|\sigma(\mathbf{Q})\mathbf{1}_d\|_2^2 = \sup_d \frac{\sigma^2(\mathbf{Q})d}{2\lambda_d} = \left(\frac{p}{1-p}\right)\frac{\sigma^2(\mathbf{Q})}{2}, \tag{6.86}$$

where recall $\lambda_d = d(1-p)/p$. The result then follows by noting the well-known duality between the spectral norm (largest singular value) and the nuclear norm and basic properties of the Fenchel dual. □

Let us remember that, as we show in Remark 2, when we compute the infimum of $\Omega_{\text{dropout}}(\mathbf{U},\mathbf{V})$ over $\mathbf{U},\mathbf{V},d$ such that $d \geq \rho(\mathbf{X})$ and $\mathbf{U}\mathbf{V}^\top = \mathbf{X}$, we get zero if the dropout retain probability θ is fixed. Differently, when $\theta = \theta(d)$ is allowed to be a function of d as in (6.73), we immediately get that the infimum of $\frac{1-\theta(d)}{\theta(d)}\inf_{\mathbf{U},\mathbf{V},d}\Omega_{\text{dropout}}(\mathbf{U},\mathbf{V})$ is not zero and, ancillary, this prevents pathological scheme like (6.66) to decrease the objective value of (3′) without really approximating $\mathbf{X}$. Differently, Proposition 5 guarantees that the adaptation of the dropout rate θ is able to constrain the regularizer in terms of a convex lower bound for it, the lower convex bound being (a scaled version) of the squared nuclear norm $\|\mathbf{X}\|_\star^2$. This enables us to retrieve a stronger connection[4] between dropout regularizer and (squared) nuclear norm, achieving a disciplined linkage between the two.

Actually, taking advantage of Proposition 5, we can provide a stronger theoretical result, which, on the one hand, establishes a direct connection between dropout for MF with variable size and squared nuclear norm regularization.

Theorem 11. *Let* $\mathbf{U}^{\text{opt}}$ *and* $\mathbf{V}^{\text{opt}}$ *the* $m \times d^{\text{opt}}$ *and* $n \times d^{\text{opt}}$ *optimal factors that achieves the global optimum of dropout for MF* (3′) *with* $\theta = \theta(d)$ *as in* (6.73)

[4] Let us clarify that such connection is not totally unexpected, even in the variable size case, since the variational form (6.19) holds when we optimize over d in addition to $\mathbf{U}$ and $\mathbf{V}$.

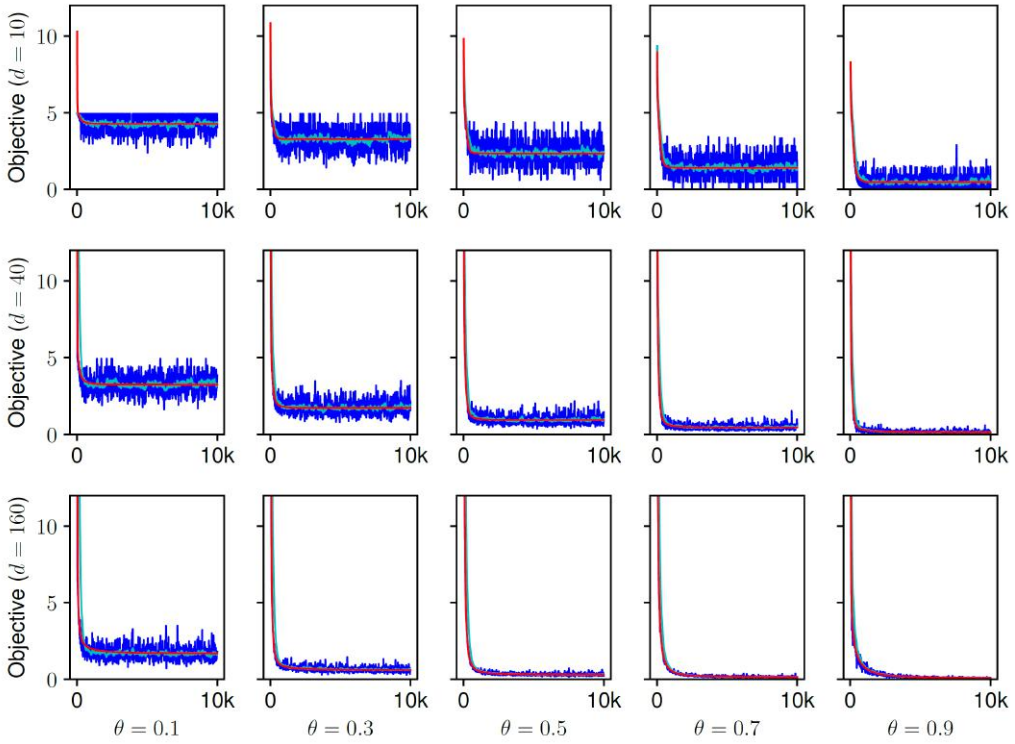


FIGURE 6.2: For $\theta \in \{0.1, 0.3, 0.5, 0.7, 0.9\}$ and $d = 160$ we compare dropout for MF (6.3) (blue) and its deterministic counterpart (red). The exponential moving average of the stochastic objective is in cyan. Best viewed in color.

for some fixed hyper-parameter p, $0 < p < 1$. *Then* $\mathbf{A}^{\text{opt}} = (\mathbf{U}^{\text{opt}}) \cdot (\mathbf{V}^{\text{opt}})^\top$ *is the global minimizer of*

$$\min_{\mathbf{A}} \left[\|\mathbf{X} - \mathbf{A}\|_F^2 + \frac{1-p}{p} \|\mathbf{A}\|_\star^2 \right], \tag{6.87}$$

which corresponds to optimizing over $\mathbf{A} \in \mathbb{R}^{m \times n}$ *the problem* (6.2) *with* $\Xi = \|\cdot\|_\star^2$ *and* $\gamma = \frac{1-p}{p}$.

Proof. In order to obtain the thesis, let us show the following result.

$$\left\langle \mathbf{Y} - \mathbf{U}\mathbf{V}^\top, \mathbf{U}\mathbf{V}^\top \right\rangle = d\frac{1-p}{p} \sum_{k=1}^{d} \|\mathbf{u}_k\|_2^2 \|\mathbf{v}_k\|_2^2 \tag{6.88}$$

Since $(\mathbf{U}, \mathbf{V})$ is a local minimizer of $f(\mathbf{U}, \mathbf{V}, d)$, then there exists $\delta > 0$ such that, for any $\epsilon > 0$, $\epsilon < \delta$, we must have

$$f(\mathbf{U}, \mathbf{V}, d) \leq f(\mathbf{U} + \epsilon\mathbf{U}, \mathbf{V} + \epsilon\mathbf{V}, d) = f((1+\epsilon)\mathbf{U}, (1+\epsilon)\mathbf{V}, d). \tag{6.89}$$

That is

$$\|\mathbf{Y} - \mathbf{U}\mathbf{V}^\top\|_F^2 + d\frac{1-p}{p} \sum_{k=1}^{d} \|\mathbf{u}_k\|_2^2 \|\mathbf{v}_k\|_2^2 \tag{6.90}$$

is upper bounded by

$$\|\mathbf{Y} - (1+\epsilon)^2\mathbf{U}\mathbf{V}^\top\|_F^2 + d\frac{1-p}{p}(1+\epsilon)^4 \sum_{k=1}^{d} \|\mathbf{u}_k\|_2^2\|\mathbf{v}_k\|_2^2. \tag{6.91}$$

Exploit the first order approximations $(1+\epsilon)^2 = 1+2\epsilon+O(\epsilon^2)$ and $(1+\epsilon)^4 = 1+4\epsilon+O(\epsilon^2)$. Since $\|\mathbf{A}+(1+\epsilon)^2\mathbf{B}\|_F^2 = \|\mathbf{A}+(1+2\epsilon)\mathbf{B}\|_F^2 + O(\epsilon^2)$,

$$\|\mathbf{Y} - \mathbf{U}\mathbf{V}^\top\|_F^2 + d\frac{1-p}{p} \sum_{k=1}^{d} \|\mathbf{u}_k\|_2^2\|\mathbf{v}_k\|_2^2 \leq \tag{6.92}$$

$$\|\mathbf{Y} - \mathbf{U}\mathbf{V}^\top - 2\epsilon\mathbf{U}\mathbf{V}^\top\|_F^2 + d\frac{1-p}{p}(1+4\epsilon) \sum_{k=1}^{d} \|\mathbf{u}_k\|_2^2\|\mathbf{v}_k\|_2^2 + O(\epsilon^2) \tag{6.93}$$

and also, by deleting $d\frac{1-p}{p}\sum_{k=1}^{d}\|\mathbf{u}_k\|_2^2\|\mathbf{v}_k\|_2^2$ and rearranging terms,

$$0 \leq \|\mathbf{Y} - \mathbf{U}\mathbf{V}^\top - 2\epsilon\mathbf{U}\mathbf{V}^\top\|_F^2 - \|\mathbf{Y} - \mathbf{U}\mathbf{V}^\top\|_F^2 + 4\epsilon d\frac{1-p}{p} \sum_{k=1}^{d} \|\mathbf{u}_k\|_2^2\|\mathbf{v}_k\|_2^2 + O(\epsilon^2). \tag{6.94}$$

Divide by $\varepsilon = 2\epsilon$,

$$0 \leq \tfrac{1}{\varepsilon}\left(\|\mathbf{Y} - \mathbf{U}\mathbf{V}^\top - \varepsilon\mathbf{U}\mathbf{V}^\top\|_F^2 - \|\mathbf{Y} - \mathbf{U}\mathbf{V}^\top\|_F^2\right) + 2d\frac{1-p}{p} \sum_{k=1}^{d} \|\mathbf{u}_k\|_2^2\|\mathbf{v}_k\|_2^2 + O(\varepsilon). \tag{6.95}$$

Take the limit as $\varepsilon \to 0$ and use the definition of one-sided directional derivative for a differentiable function h

$$\langle \nabla h(\mathbf{x}), \mathbf{d}\rangle = \nabla_{\mathbf{d}} h(\mathbf{x}) = \lim_{\varepsilon \to 0} \frac{h(\mathbf{x}+\varepsilon\mathbf{d}) - h(\mathbf{x})}{\varepsilon} \tag{6.96}$$

for $h(\cdot) = \|\mathbf{Y} - \cdot\|_F^2$, $\mathbf{x} = \mathbf{U}\mathbf{V}^\top$ and $\mathbf{d} = \mathbf{U}\mathbf{V}^\top$. Then,

$$0 \leq \left\langle \left[\nabla_{\mathbf{X}} \|\mathbf{X} - \mathbf{Y}\|_F^2\right](\mathbf{U}\mathbf{V}^\top), \mathbf{U}\mathbf{V}^\top \right\rangle + 2d\frac{1-p}{p} \sum_{k=1}^{d} \|\mathbf{u}_k\|_2^2\|\mathbf{v}_k\|_2^2 \tag{6.97}$$

$$= 2\left\langle \mathbf{U}\mathbf{V}^\top - \mathbf{Y}, \mathbf{U}\mathbf{V}^\top \right\rangle + 2d\frac{1-p}{p} \sum_{k=1}^{d} \|\mathbf{u}_k\|_2^2\|\mathbf{v}_k\|_2^2 \tag{6.98}$$

and, once the factor 2 is simplified,

$$0 \leq -\left\langle \mathbf{Y} - \mathbf{U}\mathbf{V}^\top, \mathbf{U}\mathbf{V}^\top \right\rangle + d\frac{1-p}{p} \sum_{k=1}^{d} \|\mathbf{u}_k\|_2^2\|\mathbf{v}_k\|_2^2. \tag{6.99}$$

Note that by $\epsilon > 0$ and sufficiently small, we must also have,

$$f(\mathbf{U}, \mathbf{V}, d) \leq f(\mathbf{U} - \epsilon\mathbf{U}, \mathbf{V} - \epsilon\mathbf{V}, d) = f((1-\epsilon)\mathbf{U}, (1-\epsilon)\mathbf{V}, d). \tag{6.100}$$

By applying the same steps as before, we get

$$0 \leq \frac{1}{\varepsilon}\left(\|\mathbf{Y}-\mathbf{U}\mathbf{V}^\top+\varepsilon\mathbf{U}\mathbf{V}^\top\|_F^2-\|\mathbf{Y}-\mathbf{U}\mathbf{V}^\top\|_F^2\right)-2d\tfrac{1-p}{p}\sum_{k=1}^{d}\|\mathbf{u}_k\|_2^2\|\mathbf{v}_k\|_2^2+O(\varepsilon). \tag{6.101}$$

Take the limit as $\varepsilon \to 0$ and use the definition of directional derivative (6.96).

$$0 \leq \left\langle\left[\nabla_{\mathbf{X}}\|\mathbf{X}-\mathbf{Y}\|_F^2\right](\mathbf{U}\mathbf{V}^\top),-\mathbf{U}\mathbf{V}^\top\right\rangle-2d\tfrac{1-p}{p}\sum_{k=1}^{d}\|\mathbf{u}_k\|_2^2\|\mathbf{v}_k\|_2^2 \tag{6.102}$$

$$=2\left\langle\mathbf{U}\mathbf{V}^\top-\mathbf{Y},-\mathbf{U}\mathbf{V}^\top\right\rangle-2d\tfrac{1-p}{p}\sum_{k=1}^{d}\|\mathbf{u}_k\|_2^2\|\mathbf{v}_k\|_2^2 \tag{6.103}$$

$$=2\left\langle\mathbf{Y}-\mathbf{U}\mathbf{V}^\top,\mathbf{U}\mathbf{V}^\top\right\rangle-2d\tfrac{1-p}{p}\sum_{k=1}^{d}\|\mathbf{u}_k\|_2^2\|\mathbf{v}_k\|_2^2. \tag{6.104}$$

Simplify the common factor 2, after changing the signs,

$$0 \geq -\left\langle\mathbf{Y}-\mathbf{U}\mathbf{V}^\top,\mathbf{U}\mathbf{V}^\top\right\rangle+d\tfrac{1-p}{p}\sum_{k=1}^{d}\|\mathbf{u}_k\|_2^2\|\mathbf{v}_k\|_2^2 \tag{6.105}$$

The thesis comes by combining (6.99) and (6.105). Let $(\mathbf{U},\mathbf{V},d)$ be a local minimizer of

$$\min_{\mathbf{U},\mathbf{V},d}\left[\|\mathbf{Y}-\mathbf{U}\mathbf{V}^\top\|_F^2+d\tfrac{1-p}{p}\sum_{k=1}^{d}\|\mathbf{u}_k\|_2^2\|\mathbf{v}_k\|_2^2\right] \tag{6.106}$$

and consider its convex lower bound

$$\min_{\mathbf{X}}\|\mathbf{X}-\mathbf{Y}\|_F^2+\tfrac{1-p}{p}\|\mathbf{X}\|_\star^2. \tag{6.107}$$

As a sufficient condition to achieve minimality, assume that

$$\|\mathbf{U}\mathbf{V}^\top\|_\star=\sqrt{d\sum_{k=1}^{d}\|\mathbf{u}_k\|_2^2\|\mathbf{v}_k\|_2^2} \tag{6.108}$$

and

$$\mathbf{p}^\top(\mathbf{Y}-\mathbf{U}\mathbf{V}^\top)\mathbf{q} \leq \tfrac{1-p}{p}\|\mathbf{p}\|_2\|\mathbf{q}\|_2\|\mathbf{U}\mathbf{V}^\top\|_\star \tag{6.109}$$

for any column vector $\mathbf{p}\in\mathbb{R}^m$ and $\mathbf{q}\in\mathbb{R}^n$. Then $\mathbf{U}\mathbf{V}^\top$ is a global minimizer for the convex lower bound.

Necessary condition. Assume that $\mathbf{U}\mathbf{V}^\top$ is a global minimizer for the convex lower bound. Then, (6.108) is satisfied. Thus, the first order optimality condition for

$$\min_{\mathbf{X}}\|\mathbf{X}-\mathbf{Y}\|_F^2+\lambda\|\mathbf{X}\|_\star^2 \tag{6.110}$$

is

$$0 \in \nabla_{\mathbf{X}} \|\mathbf{X}-\mathbf{Y}\|_F^2 + \partial\left(\tfrac{1-p}{p}\|\mathbf{X}\|_\star^2\right) \tag{6.111}$$

$$\Leftrightarrow 0 \in 2(\mathbf{X}-\mathbf{Y}) + 2\tfrac{1-p}{p}\|\mathbf{X}\|_\star \partial\|\mathbf{X}\|_\star \tag{6.112}$$

$$\Leftrightarrow \mathbf{Y}-\mathbf{X} \in \tfrac{1-p}{p}\|\mathbf{X}\|_\star \partial\|\mathbf{X}\|_\star \tag{6.113}$$

$$\Leftrightarrow \frac{\mathbf{Y}-\mathbf{X}}{\frac{1-p}{p}\|\mathbf{X}\|_\star} \in \partial\|\mathbf{X}\|_\star. \tag{6.114}$$

Recall that

$$\partial\|\mathbf{X}\|_\star = \left\{\mathbf{W} \in \mathbb{R}^{m\times n} \colon \langle \mathbf{X}, \mathbf{W}\rangle = \|\mathbf{X}\|_\star, \quad \mathbf{p}^\top \mathbf{W}\mathbf{q} \le \|\mathbf{p}\|_2\|\mathbf{q}\|_2 \ \forall\, (\mathbf{p},\mathbf{q})\right\}. \tag{6.115}$$

Therefore, since (6.109) easily reads as the inequality in (6.115) for

$$\mathbf{W} = \frac{\mathbf{Y}-\mathbf{U}\mathbf{V}^\top}{\frac{1-p}{p}\|\mathbf{U}\mathbf{V}^\top\|_\star}, \tag{6.116}$$

the thesis will follow if we show

$$\left\langle \mathbf{U}\mathbf{V}^\top, \frac{\mathbf{Y}-\mathbf{U}\mathbf{V}^\top}{\frac{1-p}{p}\|\mathbf{U}\mathbf{V}^\top\|_\star} \right\rangle = \|\mathbf{U}\mathbf{V}^\top\|_\star \tag{6.117}$$

or, equivalently,

$$\left\langle \mathbf{Y}-\mathbf{U}\mathbf{V}^\top, \mathbf{U}\mathbf{V}^\top \right\rangle = \tfrac{1-p}{p}\|\mathbf{U}\mathbf{V}^\top\|_\star^2. \tag{6.118}$$

We can observe that (6.118) easily follows from (6.108) if considering equation (6.88). Let $(\mathbf{U},\mathbf{V},d)$ be a global minimizer of the original problem and assume that $\mathbf{U}\mathbf{V}^\top$ satisfies the optimality conditions for the convex lower bound. Since $(\mathbf{U},\mathbf{V},d)$ is a local minimizer of the original problem, then, we have

$$\left\langle \mathbf{Y}-\mathbf{U}\mathbf{V}^\top, \mathbf{U}\mathbf{V}^\top \right\rangle = d\tfrac{1-p}{p}\sum_{k=1}^{d} \|\mathbf{u}_k\|_2^2\|\mathbf{v}_k\|_2^2. \tag{6.119}$$

At the same time, since $\mathbf{U}\mathbf{V}^\top$ satisfies the optimality conditions for the convex lower bound, we have

$$\mathbf{Y}-\mathbf{U}\mathbf{V}^\top \in \tfrac{1-p}{p}\|\mathbf{U}\mathbf{V}^\top\|_\star \partial\|\mathbf{U}\mathbf{V}^\top\|_\star. \tag{6.120}$$

Considering that

$$\partial\|\mathbf{X}\|_\star = \left\{\mathbf{W} \colon \langle \mathbf{X}, \mathbf{W}\rangle = \|\mathbf{X}\|_\star, \quad \mathbf{p}^\top \mathbf{W}\mathbf{q} \le \|\mathbf{p}\|_2\|\mathbf{q}\|_2 \ \forall\, (\mathbf{p},\mathbf{q})\right\}, \tag{6.121}$$

in particular,

$$\langle \mathbf{Y}-\mathbf{U}\mathbf{V}^\top, \mathbf{U}\mathbf{V}^\top \rangle = \tfrac{1-p}{p}\|\mathbf{U}\mathbf{V}^\top\|_\star^2. \tag{6.122}$$

Combine (6.88) and (6.122), we get

$$\|\mathbf{U}\mathbf{V}^\top\|_\star^2 = d\sum_{k=1}^{d} \|\mathbf{u}_k\|_2^2\|\mathbf{v}_k\|_2^2 \tag{6.123}$$

and (6.108) follows by square-rooting each member of (6.123). □

Theorem 11 achieves our targeted goal of exploiting dropout as a leap between matrix factorization (6.1) and approximation (6.2) problems. As we did in Section 6.1.1, thanks to the marginalization through expectation as in (3′), we are able to condensate all the stochastic suppression of columns in the factors into a fully deterministic problem (6.1) with $\Omega = \Omega_{\text{dropout}}$, and, also, the same equivalence holds when d is variable. Actually, the real reason to do that is, in such a case, we can define a variable dropout retain probability $\theta = \theta(d)$ as in (6.73) and retrieve that dropout for MF is equivalent to the optimization problem (6.87). Precisely, that "equivalence" should be interpreted as follows: the global optimum $(\mathbf{U}^{\text{opt}}, \mathbf{V}^{\text{opt}})$ of (3′) provides for free the global optimum $\mathbf{A}^{\text{opt}} = (\mathbf{U}^{\text{opt}}) \cdot (\mathbf{V}^{\text{opt}})^\top$ for (6.87).

Equation (6.126) is useful also to understand the role of the hyper-parameter p that was introduced within the definition of (6.73). In fact, the necessity of the dependence on p in $\theta(d)$ (6.73) is dictated from the exigence of allowing a variable regulation for the squared nuclear norm regularization (6.87). In fact, consistently with our goal of using dropout as a leap in between matrix factorization (6.1) and approximation (6.2), by defining the dropout retain probability θ, we are able, on the one hand, to find λ in (6.1) as $\lambda = \frac{1-\theta}{\theta}$ and, on the other hand, when $\theta(d) = \frac{p}{d-(d-1)p}$, we select γ in (6.2) to be $\gamma = \frac{1-p}{p}$. Let us observe that having dropout retain probability that depends upon hyper-parameters has been already proposed in the literature (*e.g.* [Mor+17]).

As a final remark, since the objective function of (6.87) is strictly convex, the existence and uniqueness of the global minimizer of (6.87) is guaranteed and, moreover, it can be expressed through the following closed form solution.

Theorem 12. *Let* $\mathbf{X} = \mathbf{L}\mathbf{\Sigma}\mathbf{R}^\top$ *be the singular valued decomposition of* $\mathbf{X}$*. The optimal solution* $\mathbf{A}^{\text{opt}}$ *to* (6.87) *is given by*

$$\mathbf{A}^{\text{opt}} = \mathbf{L}\mathcal{S}_\mu(\mathbf{\Sigma})\mathbf{R}^\top \tag{6.124}$$

where $\mathcal{S}_\mu(\sigma) = \max(\sigma - \mu, 0)$ *defines the shrinkage thresholding operator*[5] *[Vid+16b] applied entrywise to the singular values* $\sigma_i(\mathbf{X})$ *of* $\mathbf{X}$ *and*

$$\mu = \frac{1-p}{p + (1-p)d} \sum_{i=1}^{d} \sigma_i(\mathbf{X}) \tag{6.125}$$

where d *denotes the largest integer such that*

$$\sigma_d(\mathbf{X}) > \frac{1-p}{p + (1-p)d} \sum_{i=1}^{d} \sigma_i(\mathbf{X}). \tag{6.126}$$

Proof. Since both the nuclear norm $\|\cdot\|_\star$ and the Frobenius norm $\|\cdot\|_F$ are rotationally invariant, up to non-restrictive rotations applied to the data matrix $\mathbf{X}$, the thesis can be equivalently proved by considering the following result.

[5] For a general scalar x, one usually defines $\text{Section}_\mu(x) = \text{sgn}(x)\max(|x| - \mu, 0)$, but, here, due to the non-negativity of the singular values $\sigma > 0$, we will exploit the simplified expression $\text{Section}_\mu(\sigma) = \max(\sigma - \mu, 0)$.

Let $\mathbf{x} = [x_1, \ldots, x_r]$ a fixed vector with $x_i \geq x_{i+1} > 0$. Define μ_d as the average of the first d entries of $\mathbf{x}$ Then, the optimal solution to the optimization problem

$$\min_{\mathbf{a} \in \mathbb{R}^r} \|\mathbf{a} - \mathbf{x}\|_2^2 + \lambda \|\mathbf{a}\|_1^2 \tag{6.127}$$

is given by $\mathbf{a} = [a_1, \ldots, a_r]$ where

$$a_i = \begin{cases} x_i - \dfrac{\lambda d}{1 + \lambda d} \mu_d & i = 1, \ldots, d \\ 0 & i = d+1, \ldots, r \end{cases} \tag{6.128}$$

where d is the largest positive integer less or equal to r such that all ai given in (6.128) are positive.

In order to prove this claim, first note that the objective function is strictly convex and, hence, there is a unique global minimum. If $\lambda = 0$ the global minimizer is precisely $\mathbf{x}$, which is consistent with the formula given in the statement of the proposition. So, suppose that $\lambda > 0$. Next, notice that if $\mathbf{a} = [a_1, a_2, \ldots, a_r]$ is an optimal solution, then all a_i must be non-negative. Indeed, if say $a_1 < 0$, then the vector $[-a_1, a_2, \ldots, a_r]$ already gives a smaller objective value. Now, the first order optimality condition of our problem rewrites

$$0 \in (\mathbf{a} - \mathbf{x}) + \lambda \|\mathbf{a}\|_1 \partial \|\mathbf{a}\|_1. \tag{6.129}$$

There are two cases for each coordinate i of (6.129).

$$a_i = x_i - \lambda \|\mathbf{a}\|_1, \text{ if } a_i > 0, \text{ and } x_i = \lambda \|\mathbf{a}\|_1 \xi_i, \text{ if } a_i = 0. \tag{6.130}$$

where ξ_i in (6.130) is some number in the interval $[0, 1]$. Notice that since $x_i > 0$ for every i, the second condition in (6.130) guarantees that the global solution can not be the zero vector, otherwise $\|\mathbf{a}\|_1 = 0$ and so $x_i = 0$ for every i. Thus, suppose that exactly the first $k \geq 1$ coordinates of $\mathbf{a}$ are non-zero. Then sum the equations $a_i = x_i - \lambda k \|\mathbf{a}\|_1$ for $i = 1, \ldots, k$. We get

$$\|\mathbf{a}\|_1 = k\mu_k - \lambda k \|\mathbf{a}\|_1 \tag{6.131}$$

which gives

$$\|\mathbf{a}\|_1 = \frac{k}{1 + \lambda k} \mu_k. \tag{6.132}$$

Then (6.130) and (6.132) give

$$a_i = x_i - \frac{\lambda k}{1 + \lambda k} \mu_k > 0 \text{ for } i = 1, \ldots, k \text{ and } a_i = 0 \text{ for } i = k+1, \ldots, r. \tag{6.133}$$

Now, let d be the largest integer such that $a_i = x_i - \dfrac{\lambda d}{1 + \lambda d} \mu_d > 0$ and define the vector

$$\mathbf{v} = \left[x_1 - \frac{\lambda d}{1 + \lambda d} \mu_d, \ldots, x_d - \frac{\lambda d}{1 + \lambda d} \mu_d, \underbrace{0, \ldots, 0}_{r - d \text{ times}} \right]. \tag{6.134}$$

If $d = r$, then $\mathbf{v}$ satisfies the optimality condition (6.130) and so it is the global minimizer. So suppose that $d < r$. In that case, to show that $\mathbf{v}$ is the global

minimizer it suffices to show that

$$x_{d+1} - \frac{\lambda d}{1+\lambda d}\mu_d \leq 0. \tag{6.135}$$

since this is equivalent to saying that for any $i > d$ there exists $\xi_i \in [0,1]$ such that $x_i = \lambda\|\mathbf{v}\|_1\xi_i$ in which case $\mathbf{v}$ satisfies the optimality condition (6.130). Now by the maximality of d, we have that

$$x_{d+1} - \frac{\lambda(d+1)}{1+\lambda(d+1)}\mu_{d+1} \leq 0. \tag{6.136}$$

Equivalently, we get the following chain of inequalities

$$\left(1 - \frac{\lambda}{1+\lambda(d+1)}\right)x_{d+1} - \frac{\lambda}{1+\lambda(d+1)}\sum_{k=1}^{d} x_k \leq 0 \tag{6.137}$$

$$\frac{1+\lambda d}{1+\lambda(d+1)}x_{d+1} - \frac{\lambda}{1+\lambda(d+1)}\sum_{k=1}^{d} x_k \leq 0 \tag{6.138}$$

$$x_{d+1} - \frac{\lambda d}{1+\lambda d}\mu_d \leq 0 \tag{6.139}$$

from which we obtain the desired condition. □

The convex lower bound (6.87) to dropout for MF allows a closed-form solution in terms of the singular value decomposition of $\mathbf{X}$. While keeping the same singular vectors, the singular values are instead massaged by means of the shrinkage thresholding operator Section_μ where μ is data dependent. Moreover, in order to compute it, on needs to found d as in (6.126) before computing (6.124).

We can interpret the latter points as follows: dropout for MF with variable size is sort of acting a dimensionality reduction technique, which is very close to PCA [Vid+16b]. However, two differences arise: first, the number of principal components is not (heuristically) fixed but dropout learns it to be $d^{\text{opt}} = d$. Second, the top d singular values are not directly used for the projection, but, instead, we shrink them in a way that is adaptively induced by the data itself. Since we find this connection between dropout for MF and the sort of adaptive PCA described below, we can ultimately state that the following. Dropping out columns in the factors acts as a regularizer which promotes spectral sparsity for low-rank solutions.

6.1.4 Numetical simulations

Stochastic vs. deterministic reformulations of dropout. To demonstrate our claims experimentally, we first verify the equivalence between the stochastic (6.3) and its deterministic counterpart (6.1), in which $\Omega = \Omega_{\text{dropout}}$. To do so, we construct a synthetic data matrix $\mathbf{X}$, where $m = n = 100$, defined as the matrix product $\mathbf{X} = \mathbf{U}_0\mathbf{V}_0^\top$ where $\mathbf{U}_0, \mathbf{V}_0 \in \mathbb{R}^{100\times d}$ with $d = 10, 40, 160$. The entries of $\mathbf{U}_0$ and $\mathbf{V}_0$ were sampled from a $\mathcal{N}(0, \sigma^2)$ Gaussian distribution with standard deviation 0.1. Both the stochastic and deterministic formulations of dropout were solved by 10,000 iterations of gradient descent with diminishing

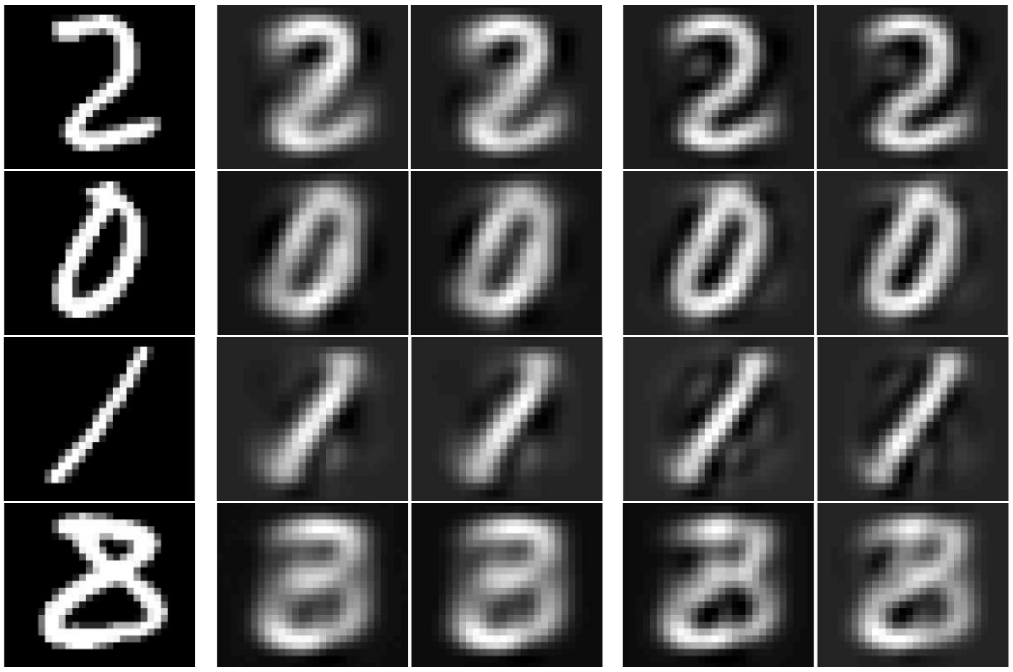


FIGURE 6.3: Experiments on MNIST dataset, whose original images are reported in the first column. For each of those, we compute dropout for MF with $\theta = 0.5$ and $\theta = 0.8$ - second and fourth columns respectively - and the two relative closed form solutions (6.124) - third and fifth columns.

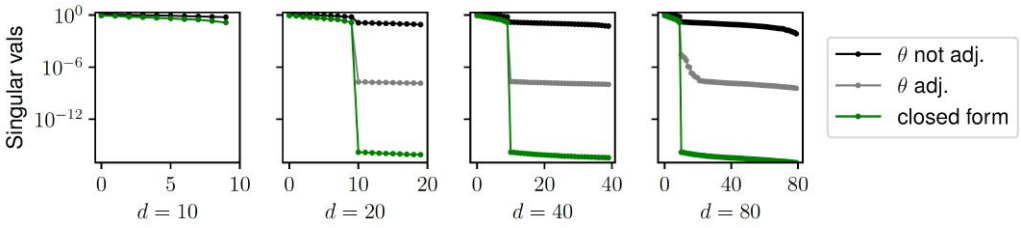


FIGURE 6.4: Singular values corresponding to the optimal solutions of the three regularization schemes considered: fixed dropout rate of $\theta = 0.9$ (black), adaptive dropout $\theta = \theta(d)$ as (6.73) with $p = 0.9$ (gray), and the nuclear-norm squared closed-form optimization as in Proposition 5 (green). Best viewed in color.

$O(1/t)$ lengths for the step size. In the stochastic setting, we approximate the objective in (6.3) and the gradient by sampling a new Bernoulli vector $\mathbf{r}$ for every iteration of Algorithm 7.

Figure 6.2 plots the objective curves for the stochastic and deterministic dropout formulations for different choices of the dropout rate $\theta = 0.1, 0.3, 0.5, 0.7, 0.9$ and factorization size $d = 10, 40, 160$. We observe that across all choices of parameters θ and d, the deterministic objective (6.1) tracks the apparent expected value that is computed in (6.3). This provides experimental evidence for the fact that the two formulations are equivalent, as predicted.

Evaluating the connections with nuclear norm. As a second experiment, we want to support the connection between Ω_{dropout} and the squared nuclear norm, in the case of a factorization with a variable size.

We constructed a synthetic dataset X consisting of a low-rank matrix combined with dense Gaussian noise. Specifically, we let $X = U_0 V_0^\top + Z_0$ where $U_0, V_0 \in \mathbb{R}^{100\times 10}$ contain entries drawn from a normal distribution $\mathcal{N}(0, \sigma^2)$, with $\sigma = 0.1$. The entries of the noise matrix Z_0 were drawn from a normal distribution with $\sigma = 0.01$. We fixed the dropout parameter $\theta = 0.9$ and run Algorithm 7.

Figure 6.4 plots the singular values for the optimal solution to each of the three problems. We observe first that without adjusting θ, dropout regularization has little effect on the rank of the solution. The smallest singular values are still relatively high and not modified significantly compared to the singular values of the original data. On the other hand, by adjusting the dropout rate based on the size of the factorization we observe that the method correctly recovers the rank of the noise-free data which also closely matches the predicted convex envelope with the nuclear-norm squared regularizer (note the log scale of the singular values). Furthermore, across the choices for d, the relative Frobenius distances between the solutions of these two methods are very small (between 10^{-6} and 10^{-2}). Taken together, our theoretical predictions and experimental results suggest that adapting the dropout rate based on the size of the factorization is critical to ensuring the effectiveness of dropout as a regularizer and in limiting the degrees of freedom of the model.

Matrix factorization meets approximation with dropout. In this Chapter, we study the process of dropping out columns of the factors $\mathbf{U}$ and $\mathbf{V}$ with which a data matrix $\mathbf{X}$ needs to be approximated in the form $\mathbf{U}\mathbf{V}^\top$. In addition to prove that this acts as a classical regularization scheme of the type (6.1), we also show that, at the optimum, the same problem is equivalent with the matrix approximation framework (6.2). As another experiment, we want to validate the quality of that approximation. In order to do this we consider MNIST training set, made of 55K images of resolution 28×28 that are vectorized and min-max normalized so that $\mathbf{X}$ has 55K rows and 784 columns.

As a first step we fix θ. Then, we applied SGD gradient descent, to compute the gradients as in Algorithm (7) with a learning rate of $\epsilon = 10^{-4}$. In order to better cope with the non-convexity of the optimization, we performed about 1000 epochs where we carried $50\times$ updates of $\mathbf{U}$ keeping $\mathbf{V}$ fixed and, conversely, $50\times$ updates of $\mathbf{V}$ while freezing $\mathbf{V}$. Due to the shallowness of the model, we did not apply any batch strategy, but gradients are computed on the whole MNIST training by using acceleration with a GTX 1080 GPU. We fixed the dimensionality of the factors to 40.

While the factors $\mathbf{U}$ and $\mathbf{V}$ are computed in the aforementioned way, we compute the matrix $\mathbf{U}\mathbf{V}^\top$, dividing by θ and we compared against the closed form solution (6.124) of (6.87). In order to do so, we first compute $\gamma = \frac{1-p}{p}$ being p obtained by solving (6.73) with respect to p while $\theta(d)$ and d are fixed. Afterwards, we compute d as in (6.126) and, finally, we compute the singular value decomposition of $\mathbf{X}$ and we invoke (6.124) (in order to avoid out-of-memory issue, the svd of $\mathbf{X}$ was computed on a computer with 256 GB of RAM using MATLAB). In Figure 6.3 we show the visual results obtained comparing the original MNIST data with their reconstruction obtained through either dropout on MF or its convex lower bound. In both cases, we used two different dropout rates $\theta = 0.5$ and $\theta = 0.8$. Visually, the two reconstructions are pretty close and this is certified analytically since the mean reconstruction error of

either dropout on MF or its convex lower bound has order of magnitude 10^{-2} and, the mean squared error between $\mathbf{U}\mathbf{V}^\top$ and (6.124) is approx. 10^{-3}.

6.2 Adaptive dropout for deep neural networks: Curriculum Dropout

Since [Kri+12b], deep neural networks have become ubiquitous in most computer vision applications. The reason is generally ascribed to the powerful hierarchical feature representations directly learnt from data, which usually outperform classical hand-crafted feature descriptors.

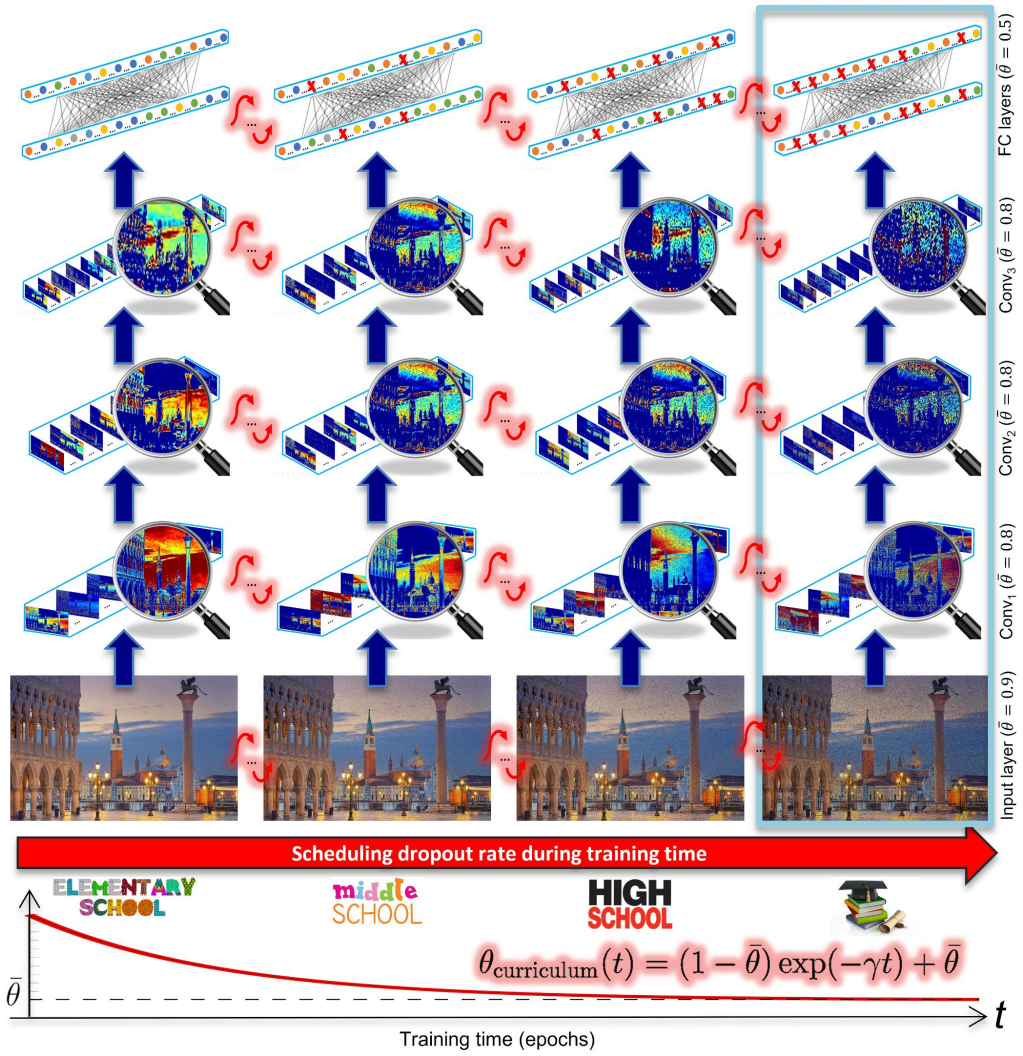
As a drawback, deep neural networks are difficult to train because non-convex optimization and intensive computations for learning the network parameters. Relying on availability of both massive data and hardware resources, the aforementioned training challenges can be empirically tackled and deep architectures can be effectively trained in an end-to-end fashion, exploiting parallel GPU computation.

However, *overfitting* remains an issue. Indeed, such a gigantic number of parameters is likely to produce weights that are so specialized to the training examples that the network's generalization capability may be extremely poor.

The seminal work of [Hin+12] argues that overfitting occurs as the result of excessive co-adaptation of feature detectors which manage to perfectly explain the training data. This leads to overcomplicated models which unsatisfactory fit unseen testing data points. To address this issue, the Dropout algorithm was proposed and investigated in [Hin+12; Sri+14] and is nowadays extensively used in training neural networks. The method consists in randomly suppressing neurons during training according to the values r sampled from a Bernoulli distribution. More specifically, if $\mathrm{r} = 1$ that unit is kept unchanged, while if r=0 the unit is suppressed. The effect of suppressing a neuron is that the value of its output is set to zero during the forward pass of training, and its weights are not updated during the backward pass. One one forward-backward pass is completed, a new sample of r is drawn from each neuron, and another forward-backward pass is done and so on till convergence. At testing time, no neuron is suppressed and all activations are modulated by the mean value of the Bernoulli distribution. The resulting model is in fact often interpreted as an average of multiple models, and it is argued that this improves its generalization ability [Hin+12; Sri+14].

Leveraging on the Dropout idea, many works have proposed variations of the original strategy [JF16; Ren+14; WG15; WM13; Bay+13; LBY16]. However, it is still unclear which variation improves the most with respect to the original dropout formulation [Hin+12; Sri+14]. In many works (such as [Ren+14]) there is no real theoretical justification of the proposed approach other than favorable empirical results. Therefore, providing a sound justification still remains an open challenge. In addition, the lack of publicly available implementations (*e.g.* , [LBY16]) make fair comparisons problematic.

The point of departure of our work is the intuition that the excessive co-adaptation of feature detectors, which leads to overfitting, are very unlikely



.

FIGURE 6.5: From left to right, we represent training time. At the beginning, due to the random initialization of the network's weights, we conjecture that there is no need to perform regularization. That is because parameters have not been optimized enough in order for the net to memorize the training set. Such situation is pop up at some point during training: that's why we propose to introduce regularization gradually by scheduling the dropout probability with a negative exponential function. In practice, we can show that our choice induces a curriculum learning [Ben+09] in which we implicitly generate some data partition where, the original dataset is corrupted with an increasing level of Bernoulli random noise, being the same kind of noise is applied to higher layers of the network. Differently from [Ben+09], we are not asked to manually compute the partitions and to train on each of those separately, but, differently, our scheduled retain probability does everything in an end-to-end manner

to occur in the early epochs of training. Thus, Dropout seems unnecessary at the beginning of training. Inspired by these considerations, in this work we propose to dynamically increase the number of units that are suppressed

as a function of the number of gradient updates. Specifically, we introduce a generalization of the dropout scheme consisting of a temporal scheduling - a *curriculum* - for the expected number of suppressed units. By adapting in time the parameter of the Bernoulli distribution used for sampling, we smoothly increase the suppression rate as training evolves, thereby improving the generalization of the model.

In summary, we provide the following main contributions in order to improve classical dropout training for deep (convolutional) neural networks.

1. We address the problem of overfitting in deep neural networks by proposing a novel regularization strategy called Curriculum Dropout that dynamically increases the expected number of suppressed units in order to improve the generalization ability of the model.

2. We draw connections between the original dropout framework [Hin+12; Sri+14] with regularization theory [Evg+00] and curriculum learning [Ben+09]. This provides an improved justification of (Curriculum) Dropout training, relating it to existing machine learning methods.

3. We complement our foundational analysis with a broad experimental validation, where we compare our Curriculum Dropout versus the original one [Hin+12; Sri+14] and anti-Curriculum [Ren+14] paradigms, for (convolutional) neural network-based image classification. We evaluate the performance on standard datasets (MNIST, SVHN [Net+11], CIFAR-10/100 [KH09], Caltech-101/256 [FF+04; Gri+07]). As the results certify, the proposed method generally achieves a superior classification performance.

The remaining of this Section is outlined as follows. Our temporal scheduling for the train probability is presented in Section 6.2.1 and Section 6.2.2, providing foundational interpretations. The experimental evaluation is carried out in Section 6.2.3. Conclusions and future work are presented in Section 4.5.

6.2.1 A Time Scheduling for the Dropout Rate

Deep Neural Networks display co-adaptations between units in terms of concurrent activations of highly organized clusters of neurons. During training, the latter specialize themselves in detecting certain details of the image to be classified, as shown by Zeiler and Fergus [ZF14]. They visualize the high sensitivity of certain filters in different layers in detecting dogs, people's faces, wheels and more general ordered geometrical patterns [ZF14, Fig. 2]. Moreover, such co-adaptations are highly generalizable across different datasets as proved by Torralba's work [Zho+14]. Indeed, the filter responses provided in the AlexNet within *conv1*, *pool2/5* and *fc7* layers are very similar [Zho+14, Fig. 5], despite the images used for the training are very different: objects from ImageNet versus scenes from Places datasets.

These arguments support the existence of some *positive* co-adaptations between neurons in the network. Nevertheless, as soon as the training keeps going, some co-adaptations can also be *negative* if excessively specific of the training images exploited for updating the gradients. Consequently, exaggerated co-adaptations between neurons weaken the network generalization capability,

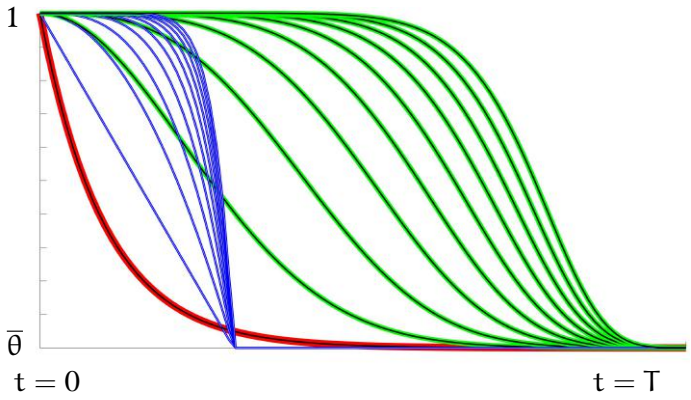


FIGURE 6.6: Curriculum functions. Eq. (6.140) (red), polynomial (blue) and exponential (green).

ultimately resulting in overfitting. To prevent it, Dropout [Hin+12; Sri+14] precisely contrasts those negative co-adaptations.

The latter can be removed by randomly suppressing neurons of the architecture, restoring an improved situation where the neurons are more "independent". This empirically reflects into a better generalization capability [Hin+12; Sri+14].

Network training is a dynamic process. Despite the previous interpretation is totally sound, the original Dropout algorithm cannot precisely accommodate for it. Indeed, the suppression of a neuron in a given layer is modeled by a Bernoulli(θ) random variable[6], $0 < \theta \leq 1$. Employing such distribution is very natural, since it statistically models binary activation/inhibition processes. In spite of that, it seems suboptimal that θ should be *fixed* during the whole training stage. With this operative choice, [Hin+12; Sri+14] is actually treating the negative co-adaptations phenomena as uniformly distributed during the whole training time.

Differently, our intuition is that, *at the beginning of the training, if any co-adaptation between units is displayed, this should be preserved* as positively representing the self-organization of the network parameters towards their optimal configuration.

We can understand this by considering the random initialization of the network's weights. They are statistically independent and actually not co-adapted at all. Also, it is quite unnatural for a neural network with random weights to overfit the data. On the other hand, the risk of overdone co-adaptations increases as the training proceeds since the loss minimization can achieve a small objective value by overcomplicating the hierarchical representation learnt from data. This implies that *overfitting caused by excessive co-adaptations appears only after a while.*

Since a fixed parameter θ is not able to handle increasing levels of negative co-adaptations, in this work, we tackle this issue by proposing a temporal dependent $\theta(t)$ parameter. Here, t denotes the training time, measured in gradient updates $t \in \{0, 1, 2, \dots\}$. Since $\theta(t)$ models the probability for a given

[6] To avoid confusion in our notation, please note that θ is the equivalent of p in [Hin+12; Sri+14; Wag+13], i.e the probability of *retaining* a neuron.

neuron to be retained, $D \cdot \theta(t)$ will count the average number of units which remain active over the total number D in a given layer. Intuitively, such quantity must be higher for the first gradient updates, then starting decreasing as soon as the training gears. In the late stages of training, such decrease should be stopped. We thus constrain $\theta(t)$ to be $\theta(t) \geq \overline{\theta}$ for any t, where $\overline{\theta}$ is a limit value, to be taken as $0.5 \leq \overline{\theta} \leq 0.9$ as prescribed by the original dropout scheme [Sri+14, Section A.4] (the higher the layer hierarchy, the lower the retain probability).

Inspired by the previous considerations, we propose the following definition for a ***curriculum function*** $\theta(t)$ aimed at improving dropout training (as it will become clear in section 6.2.2, from now on we will often use the terms *curriculum* and *scheduling* interchangeably).

Definition 3. *Any function* $t \mapsto \theta(t)$ *such that* $\theta(0) = 1$ *and* $\lim_{t\to\infty} \theta(t) \searrow \overline{\theta}$ *is said to be a curriculum function to generalize the original dropout [Hin+12; Sri+14] formulation with retain probability* $\overline{\theta}$.

Starting from the initial condition $\theta(0) = 1$ where no unit suppression is performed, dropout is gradually introduced in a way that $\theta(t) \geq \overline{\theta}$ for any t. Eventually (*i.e.* when t is big enough), the convergence $\theta(t) \to \overline{\theta}$ models the fact that we retrieve the original formulation of [Hin+12; Sri+14] as a particular case of our curriculum.

Among the functions as in Def. 3, in our work we fix

$$\theta_{\text{curriculum}}(t) = (1 - \overline{\theta}) \exp(-\gamma t) + \overline{\theta},\ \gamma > 0 \tag{6.140}$$

By considering Figure 6.6, we can provide intuitive and straightforward motivations regarding our choice.

The blue curves in Fig. 6.6 are polynomials of increasing degree $\delta = \{1, \dots, 10\}$ (left to right). Despite actually fulfilling the initial constraint $\theta(0) = 1$, they have to be manually thresholded to impose $\theta(t) \to \overline{\theta}$ when $t \to \infty$. This introduces two more (undesired) parameters (δ and the threshold) with respect to [Hin+12; Sri+14], where the only quantity to be selected is $\overline{\theta}$.

The very same argument discourages the replacement of the variable t by t^{α} in (6.140), (green curves in Fig. 6.6, $\alpha = \{2, \dots, 10\}$, left to right). Moreover, by evaluating the area under the curve, we can intuitively measure how aggressively the green curves behave while delaying the dropping out scheme they eventually converge to (as $\theta(t) \to \overline{\theta}$). Precisely, that convergence is faster while moving to the green curves more on the left, being the fastest one achieved by our scheduling function (6.140) (red curve, Fig. 6.6).

Actually, one can still argue that the parameter $\gamma > 0$ is annoying since it requires cross validation. This is not necessary: in fact, γ can actually be *fixed* according to the following heuristics. Despite Def. 3 considers the limit of $\theta(t)$ for $t \to \infty$, such condition has to be operatively replaced by $t \approx T$, being T the total number of gradient updates needed for optimization. It is thus totally reasonable to assume that the order of magnitude of T is a priori known and fixed to be some power of 10 such as $10^4, 10^5$. Therefore, for a curriculum function as in Def. 3, we are interested in furthermore imposing $\theta(t) \approx \overline{\theta}$ when $t \approx T$. Actually, a rule of thumb such as

$$\gamma = 10/T \tag{6.141}$$

implies $|\theta_{\text{curriculum}}(T) - \bar{\theta}| < 10^{-4}$ and was used for all the experiments in Section 6.2.3. Additionally, from Figure 6.6, we can grab some intuitions about the fact that the asymptotic convergence to $\bar{\theta}$ is indeed realized for a quite consistent part of the training and well before $t \approx T$. This means that during a big portion of the training, we are actually dropping out neurons as prescribed in [Hin+12; Sri+14], addressing the overfitting issue. In addition to these arguments, we will provide complementary insights on our scheduled implementation for dropout training.

Smarter initialization for the network weights. The problem of optimizing deep neural networks is non-convex due to the non-linearities (ReLUs) and pooling steps. In spite of that, a few theoretical papers have investigated this issue under a sound mathematical perspective. For instance, under mild assumptions, Haeffele and Vidal [HV15] derive sufficient conditions to ensure that a local minimum is also a global one to guarantee that the former can be found when starting from *any* initialization. Actually, the same theory presented in [HV15] cannot be straightforwardly applied to the dropout case due to the pure deterministic framework of the theoretical analysis that is carried out. Therefore, it is still an open question whether all initializations are equivalent for the sake of a dropout training and, if not, which ones are preferable. Far from providing any theoretical insight in this flavor, we posit that Curriculum Dropout can be interpreted as a smarter initialization. Indeed, we implement a soft transition between a classical dropout-free training of a network versus the dropout one [Hin+12; Sri+14]. Under this perspective, our curriculum seems equivalent to performing dropout training of a network whose weights have already been slightly optimized, evidently resulting in a better initialization for them.

Curriculum Dropout as adaptive regularization. Several connections have been established between Dropout and model training with noise addition [Bis95; Rif+11; Wag+13; Wan+13a; Sri+14; ZZ15]. The common trend discovered is that when an unregularized loss function is optimized to fit artificially corrupted data, this is actually *equivalent* to minimize the same loss augmented by a data dependent penalizing term. In both [Wag+13, Table 2.] for linear/logistic regression and [Sri+14, Section 9.1] for least squares, it is proved that Dropout induces a regularizer which is scaled by $\theta(1-\theta)$.

Theorem 13 (Dropout - least squares [Sri+14]). *Assume a least square fitting*

$$\min_{\mathbf{w}\in\mathbb{R}^d} \sum_{i=1}^{N} (y_i - \mathbf{w}^\top \mathbf{x}_i)^2 = \min_{\mathbf{w}\in\mathbb{R}^d} \|\mathbf{y} - \mathbf{X}\mathbf{w}\|_2^2 \tag{6.142}$$

to fit a linear model to explain the data, being $\mathbf{y} = [y_1, \dots, y_N]^\top$ *and* $\mathbf{X}$ *the* $N \times d$ *matrix, whose* i*-th row* $[X_{i1}, \dots, X_{id}] = \mathbf{x}_i^\top$. *According to [Hin+12; Sri+14], the dropout problem on the least squares fitting* (6.142)*, rewrites*

$$\min_{\mathbf{w}\in\mathbb{R}^d} \mathbb{E}_{\mathbf{r}} \left[\sum_{i=1}^{N} (y_i - \mathbf{w}^\top (\mathbf{r} \odot \mathbf{x}_i))^2 \right] = \min_{\mathbf{w}\in\mathbb{R}^d} \mathbb{E}_{\mathbf{r}} \|\mathbf{y} - \mathbf{X}\text{diag}(\mathbf{r})\mathbf{w}\|_2^2, \tag{6.143}$$

being $\mathbf{r} = [r_1, \dots, r_d]$ *and* $r_j \sim \text{Bernoulli}(\theta)$ *i.i.d. It results*

$$\mathbb{E}_{\mathbf{r}}\|\mathbf{y} - \mathbf{X}\text{diag}(\mathbf{r})\mathbf{w}\|_2^2 = \theta(1-\theta)\|\text{diag}(\mathbf{X}^\top\mathbf{X})^{1/2}\mathbf{w}\|_2^2 + \|\mathbf{y} - \theta\mathbf{X}\mathbf{w}\|_2^2 \tag{6.144}$$

$$= \theta(1-\theta)\|\mathbf{w}\|^2_{\text{diag}(\mathbf{X}^\top\mathbf{X})} + \|\mathbf{y} - \theta\mathbf{X}\mathbf{w}\|_2^2 \tag{6.145}$$

where the same Euclidean loss function used within the expectation is now augmented with a squared data-dependent norm induced by the matrix $\text{diag}(\mathbf{X}^\top\mathbf{X})$*, the latter being scaled by a factor* $\theta(1-\theta)$*.*

Proof. Using the definition of the Euclidean norm and the linearity of the expected value, we get

$$\mathbb{E}_{\mathbf{r}}\|\mathbf{y}-\mathbf{X}\text{diag}(\mathbf{r})\mathbf{w}\|_2^2 = \mathbb{E}_{\mathbf{r}}\left[\sum_{i=1}^{N}(y_i - \mathbf{w}^\top(\mathbf{r}\odot\mathbf{x}_i))^2\right] = \sum_{i=1}^{N}\mathbb{E}_{\mathbf{r}}\left[(y_i - \mathbf{w}^\top(\mathbf{r}\odot\mathbf{x}_i))^2\right]. \tag{6.146}$$

Apply the bias-variance decomposition $\mathbb{E}[Z^2] = \mathbb{V}[Z] + \mathbb{E}[Z]^2$, holding for any scalar random variable Z.

$$\mathbb{E}_{\mathbf{r}}\|\mathbf{y} - \mathbf{X}\text{diag}(\mathbf{r})\mathbf{w}\|_2^2 = \sum_{i=1}^{N}\left[\mathbb{V}_{\mathbf{r}}\left[y_i - \mathbf{w}^\top(\mathbf{r}\odot\mathbf{x}_i)\right] + \left(\mathbb{E}_{\mathbf{r}}\left[y_i - \mathbf{w}^\top(\mathbf{r}\odot\mathbf{x}_i)\right]\right)^2\right]. \tag{6.147}$$

The operator $\mathbb{V}_{\mathbf{r}}$ is invariant to deterministic translations. Therefore,

$$\mathbb{E}_{\mathbf{r}}\|\mathbf{y} - \mathbf{X}\text{diag}(\mathbf{r})\mathbf{w}\|_2^2 = \sum_{i=1}^{N}\left[\mathbb{V}_{\mathbf{r}}\left[-\mathbf{w}^\top(\mathbf{r}\odot\mathbf{x}_i)\right] + \left(\mathbb{E}_{\mathbf{r}}\left[y_i - \mathbf{w}^\top(\mathbf{r}\odot\mathbf{x}_i)\right]\right)^2\right]. \tag{6.148}$$

Once expanded the product $\mathbf{w}^\top(\mathbf{r}\odot\mathbf{x}_i)$ in components, we get

$$\mathbb{E}_{\mathbf{r}}\|\mathbf{y} - \mathbf{X}\text{diag}(\mathbf{r})\mathbf{w}\|_2^2 = \sum_{i=1}^{N}\left[\mathbb{V}_{\mathbf{r}}\left[-\sum_{j=1}^{d}w_j r_j X_{ij}\right] + \left(\mathbb{E}_{\mathbf{r}}\left[y_i - \sum_{j=1}^{d}w_j r_j X_{ij}\right]\right)^2\right] \tag{6.149}$$

For each i-th term of the summation, use the properties of variance and expected values with respect to linear combinations of independent random variables. This yields

$$\mathbb{E}_{\mathbf{r}}\|\mathbf{y} - \mathbf{X}\text{diag}(\mathbf{r})\mathbf{w}\|_2^2 = \sum_{i=1}^{N}\left[\sum_{j=1}^{d}w_j^2 X_{ij}^2\mathbb{V}_{\mathbf{r}}[r_j] + \left(y_i - \sum_{j=1}^{d}w_j X_{ij}\mathbb{E}_{\mathbf{r}}[r_j]\right)^2\right] \tag{6.150}$$

$$= \sum_{i=1}^{N}\left[\sum_{j=1}^{d}w_j^2 X_{ij}^2\cdot\theta(1-\theta) + \left(y_i - \sum_{j=1}^{d}w_j X_{ij}\cdot\theta\right)^2\right], \tag{6.151}$$

being the latter equality a direct consequence of the formulæ for the variance and the expected value for a Bernoulli(θ) distribution. Therefore, by highlighting the terms $\theta(1-\theta)$ and θ in front of the relative summations, we get

$$\mathbb{E}_{\mathbf{r}}\|\mathbf{y}-\mathbf{X}\text{diag}(\mathbf{r})\mathbf{w}\|_2^2 = \theta(1-\theta)\sum_{i=1}^{N}\sum_{j=1}^{d} w_j^2 X_{ij}^2 + \sum_{i=1}^{N}\left(y_i - \theta\sum_{j=1}^{d} X_{ij}w_j\right)^2. \tag{6.152}$$

By using the definition of Euclidean norm,

$$\mathbb{E}_{\mathbf{r}}\|\mathbf{y}-\mathbf{X}\text{diag}(\mathbf{r})\mathbf{w}\|_2^2 = \theta(1-\theta)\sum_{i=1}^{N}\sum_{j=1}^{d} w_j^2 X_{ij}^2 + \|\mathbf{y}-\theta\mathbf{X}\mathbf{w}\|_2^2 \tag{6.153}$$

Let us consider the first addend of (6.153) separately. By rearranging the summing ordering and replacing X_{ij}^2 with two identical copies of X_{ij}, we get

$$\sum_{i=1}^{N}\sum_{j=1}^{d} w_j^2 X_{ij}^2 = \sum_{j=1}^{d} w_j^2\left(\sum_{i=1}^{N} X_{ij}^2\right) = \sum_{j=1}^{d} w_j^2\left(\sum_{i=1}^{N} X_{ij}X_{ij}\right) \tag{6.154}$$

$$= \sum_{j=1}^{d} w_j^2\left(\sum_{i=1}^{N} (\mathbf{X})_{ji}^{\top} X_{ij}\right) = \sum_{j=1}^{d} w_j^2 [\text{diag}(\mathbf{X}^{\top}\mathbf{X})]_{jj} \tag{6.155}$$

where we have exploited the transposition and the row-by-column product definitions. By squaring and square-rooting the second factor in the summation we obtain

$$\sum_{i=1}^{N}\sum_{j=1}^{d} w_j^2 X_{ij}^2 = \sum_{j=1}^{d} w_j^2([\text{diag}(\mathbf{X}^{\top}\mathbf{X})]_{jj}^{1/2})^2. \tag{6.156}$$

By noticing that the square-root of a diagonal matrix is a diagonal matrix whose entries are the square roots of the original entries, we obtain

$$\sum_{i=1}^{N}\sum_{j=1}^{d} w_j^2 X_{ij}^2 = \sum_{j=1}^{d} w_j^2([\text{diag}(\mathbf{X}^{\top}\mathbf{X})^{1/2}]_{jj})^2 = \sum_{j=1}^{d}(w_j[\text{diag}(\mathbf{X}^{\top}\mathbf{X})^{1/2}]_{jj})^2. \tag{6.157}$$

Apply the definition of row-by-column matrix product between a diagonal matrix and a vector.

$$\sum_{i=1}^{N}\sum_{j=1}^{d} w_j^2 X_{ij}^2 = \sum_{j=1}^{d}([\text{diag}(\mathbf{X}^{\top}\mathbf{X})^{1/2}\mathbf{w}]_j)^2 \tag{6.158}$$

$$= \|\text{diag}(\mathbf{X}^{\top}\mathbf{X})^{1/2}\mathbf{w}\|_F^2, \tag{6.159}$$

where, in (6.159), we used the definition of Frobenius norm. Replacing (6.159) in (6.153), leads to to prove (6.144).

In order to elicit (6.145), it is enough to notice that (6.156) can be rewritten as

$$\sum_{j=1}^{d} w_j^2 [\text{diag}(\mathbf{X}^{\top}\mathbf{X})]_{jj} = \sum_{j=1}^{d} w_j [\text{diag}(\mathbf{X}^{\top}\mathbf{X})]_{jj} w_j = \mathbf{w}^{\top}\text{diag}(\mathbf{X}^{\top}\mathbf{X})\mathbf{w}, \tag{6.160}$$

being the last term equivalent to $\|\mathbf{w}\|_{\text{diag}(\mathbf{X}^\top\mathbf{X})}$, by exploiting the definition of norm induced by a symmetric and positive definite matrix. Therefore, (6.160), once plugged into (6.156), leads to prove (6.145). This completes the proof. □

We can apply the identical analysis to the case of a deep neural network. In such a case, the input data matrix $\mathbf{X}$ is processed across subsequences of ℓ linear layers (represented by weights $\mathbf{W}^{(\ell)}$), with intermediate gating functions, pooling and feature normalization steps. Despite the latter non-linearities, since the values sampled from a Bernoulli are always either 0 or 1, it is enough to enumerate all the possible binary combinations of activations/inhibitions, accounting for the probability of their occurrence. This allows to retrieve (a different regularization term but) the same weighting factor $\theta(1-\theta)$.

When $\theta = \overline{\theta}$, the impact of the regularization is just *fixed*, therefore rising potential over- and under-fitting issues [Evg+00]. But, for $\theta = \theta_{\text{curriculum}}(t)$, when t is small, the regularizer is set to zero ($\theta_{\text{curriculum}}(0) = 1$) and we *do not* perform any regularization at all. Indeed, the latter is simply not necessary: the network weights still have values which are close to their random and statistically independent initialization. Hence, overfitting is unlikely to occur at early training steps. Differently, we should expect it to occur as soon as training proceeds: by using (6.140), the regularizer is now weighted by

$$\theta_{\text{curriculum}}(t)(1-\theta_{\text{curriculum}}(t)), \tag{6.161}$$

which is an increasing function of t. Therefore, the more the gradient updates t, the heavier the effect of the regularization. This is the reason why overfitting is better tackled by the proposed curriculum. Despite the overall idea of an adaptive selection of parameters is not novel for either regularization theory [HR94; Cra+09; Boy+11; Sol+13; CM15] or tuning of network hyper-parameters (e.g. learning rate, [Cag+17]), to the best of our knowledge, this is the first time that this concept of time-adaptive regularization is applied to deep neural networks.

Compendium. Let us conclude with some general comments. We posit that there is no overfitting at the beginning of the network training. Therefore, differently from [Hin+12; Sri+14], we allow for a scheduled retain probability $\theta(t)$ which gradually drops neurons out. Among other plausible curriculum functions as in Def. 3, the proposed choice (6.140) introduces no additional parameter to be tuned and implicitly provides a smarter weight initialization for dropout training.

The superiority of (6.140) also relates to i) the smoothly increasingly amount of units suppressed and ii) the soft adaptive regularization performed to contrast overfitting.

Throughout these interpretations, we can retrieve a common idea of smoothly changing difficulty of the training which is applied to the network. This fact can be better understood by finding the connections with Curriculum Learning [Ben+09], as we explain in the next section.

6.2.2 Curriculum Learning and Curriculum Dropout

For the sake of clarity, let us remind the concept of curriculum learning [Ben+09]. Within a classical machine learning algorithm, all training examples are presented to the model in an unordered manner, frequently applying a random shuffling. Actually, this is very different from what happens for the human training process, that is education. Indeed, the latter is highly structured so that the level of difficulty of the concepts to learn is proportional to the *age* of the people, managing easier knowledge when babies and harder when adults. This "start small" paradigm will likely guide the learning process [Ben+09].

Following the same intuition, [Ben+09] proposes to subdivide the training examples based on their difficulty. Then, the learning is configured so that easier examples come first, eventually complicating them and processing the hardest ones at the end of the training. This concept is formalized by introducing a learning time $\lambda \in [0,1]$, so that training begins at $\lambda = 0$ and ends at $\lambda = 1$. At time λ, $Q_\lambda(z)$ denotes the distribution which a training example z is drawn from. The notion of curriculum learning is formalized requiring that Q_λ ensures a sampling of examples z which are easier than the ones sampled from $Q_{\lambda+\varepsilon}$, $\varepsilon > 0$. Mathematically, this is formalized by assuming

$$Q_\lambda(z) \propto W_\lambda(z) P(z). \tag{6.162}$$

In (6.162), $P(z)$ is the target training distribution, accounting for all examples, both easy and hard ones. The sampling from P is corrected by the factor $0 \leq W_\lambda(z) \leq 1$ for any λ and z. The interpretation for $W_\lambda(z)$ is the measure of the difficulty of the training example z. The maximal complexity for a training example is fixed to 1 and reached at the end of the training, *i.e.* $W_1(z) = 1$, *i.e.* $Q_1(z) = P(z)$. The relationship

$$W_\lambda(z) \leq W_{\lambda+\varepsilon}(z) \tag{6.163}$$

represents the increased complexity of training examples from instant λ to $\lambda+\varepsilon$. Moreover, the weights $W_\lambda(z)$ must be chosen in such a way that

$$H(Q_\lambda) < H(Q_{\lambda+\varepsilon}), \tag{6.164}$$

where Shannon's entropy $H(Q_\lambda)$ models the fact that the quantity of information exploited by the model during training increases with respect to λ.

In order to prove that our scheduled dropout fulfills this definition, for simplicity, we will consider it as applied to the input layer only. This is not restrictive since the same considerations apply to any intermediate layer, by considering that each layer trains the feature representation used as input by the subsequent one.

As the images exploited for training, consider the partitions in the dataset including all the (original) clean data and all the possible ways of corrupting them through the Bernoulli multiplicative noise (see Fig. 6.5). Let π denote the probability of sampling an uncorrupted d-dimensional image within an image dataset (nothing more than a uniform distribution over the available training examples). Let us fix the gradient update t. The case of sampling a dropped-out z is equivalent to sampling the corresponding uncorrupted image z_0 from π and then overlapping it with a binary mask b (of size d), where each entry

of b is zero with probability $1-\theta(t)$. By mapping b to the number i of its zeros,

$$\mathbb{P}[z] = \mathbb{P}[z_0, i] = \binom{d}{i}(1-\theta(t))^i\theta(t)^{d-i} \cdot \pi(z_0). \tag{6.165}$$

Indeed, $(1-\theta(t))^i\theta(t)^{d-i}$ is the probability of sampling *one* binary mask b with i zeros and $\binom{d}{i}$ accounts for all the possible combinations. Re-parameterizing the training time $t = \lambda T$, we get

$$Q_\lambda(z) = \binom{d}{i}(1-\theta(\lambda T))^i\theta(\lambda T)^{d-i} \cdot \pi(z_0). \tag{6.166}$$

By defining $P(z) = Q_1(z)$ and

$$W_\lambda(z) = \frac{1}{P(z)}\binom{d}{i}(1-\theta(\lambda T))^i\theta(\lambda T)^{d-i} \cdot \pi(z_0), \tag{6.167}$$

we can easily prove that the definition in [Ben+09] is fulfilled by the choice (6.166) for curriculum learning distribution $Q_\lambda(z)$.

Theorem 14. *Curriculum dropout scheme* (6.140) *induces a curriculum learning distribution.*

Proof. Let us denote $\mathcal{Z}_0$ the original dataset and assume to sample from it a d-dimensional image z_0 according to a distribution π. Clearly, the natural choice for π will be a uniform distribution. Moreover, here, we measure the dimensionality d of image by means of the total number of pixels.

While dropping out units in the input layer (*i.e.* pixels in z_0), we augment $\mathcal{Z}_0$ by adding all images in $\mathcal{Z}_0$ with *one* pixel se to zero (colored in black) and also all images in $\mathcal{Z}_0$ with *two* pixel se to zero and so on. This creates the dataset $\mathcal{Z}$, effectively used for dropout training, where any image $z \in \mathcal{Z}$ is obtained from an image $z_0 \in \mathcal{Z}_0$ by corrupting it through multiplicative Bernoulli noise. Equivalently, we can think about entrywise multiplying z_0 with a binary mask b. Therefore, we get

$$\mathbb{P}[\text{sampling } z] = \mathbb{P}[\text{sampling } z_0] \cdot \mathbb{P}[\text{sampling b}] = \pi(z_0) \cdot \mathbb{P}[\text{sampling b}]$$

In other words, any dropped out image z is uniquely determined by the original image z_0 and the binary mask b. One way to characterize that masks is by counting i, that is the number of zero entries of b. That leads to

$$\mathbb{P}[\text{sampling } z] = \pi(z_0) \cdot \binom{d}{i}(1-\theta)^i\theta^{d-i} \tag{6.168}$$

since b has entries set to zero (each realized with probability $1-\theta$) and the remaining set to one. The latter, being $d-i$ in total, are realized in correspondence of a success for the Bernoulli(θ) variable: therefore we obtain the term θ^{d-i}.

Let us introduce our curriculum function $\theta(t) = (1-\bar{\theta})\exp(-\gamma t) + \bar{\theta}$ (we will omit the pedix "curriculum" for notational simplicity). Let us re-parametrize $t = \lambda T$ such that the training time (measured from 0 to the total number T of gradients updates) spans the range $[0,1]$, starting at time $\lambda = 0$ and ending

at time $\lambda = 1$. Therefore, by modifying (6.168), we introduce the following curriculum learning distribution

$$Q_\lambda(z) = \pi(z_0) \cdot \binom{d}{i} (1 - \theta(\lambda T))^i \theta(\lambda T)^{d-i}. \tag{6.169}$$

Let us define

$$P(z) = Q_1(z). \tag{6.170}$$

When re-parametrizing $Q_\lambda(z) = Q_\lambda(z_0, i)$, we get a mixed distribution (discrete with respect to i and continous with respect to z_0). Hence,

$$\int Q_\lambda(z)dz = \int_{\mathcal{Z}_0} \pi(z_0)dz_0 \cdot \sum_{i=0}^{d} \binom{d}{i} (1 - \theta(\lambda T))^i \theta(\lambda T)^{d-i} = 1 \tag{6.171}$$

because π is a normalized over its support $\mathcal{Z}_0$ and because the second factor equals one thanks to the Binomial Theorem.

If we compute the entropy of Q_λ, we obtain

$$H(Q_\lambda) = H(\text{Binomial}(d,\theta(\lambda T))) \cdot H(\pi), \tag{6.172}$$

being

$$H(\text{Binomial}(d,\theta(\lambda T))) = \frac{1}{2} \log[2\pi e d \cdot \theta(\lambda T)(1 - \theta(\lambda T)] + O\left(\frac{1}{d}\right) \tag{6.173}$$

a strictly increasing function of λ. To see that, notice that it is enough to prove that $\theta(\lambda T)(1 - \theta(\lambda T))$ is increasing as a function of λ. But, this is true since composition of the composition of strictly decreasing functions is strictly increasing. Precisely, the two functions to be composed are $\theta(\lambda T)$ and $f(x) = x(1 - x)$, both of them strictly decreasing. Indeed,

$$\theta'(\lambda T) = -\gamma T(1 - \overline{\theta}) \exp(-\gamma \lambda T) < 0$$

for any λ and

$$f'(x) = 1 - 2x < 0$$

since we evaluate $f(\theta(\lambda T))$ and $\theta(\lambda T) > \overline{\theta} \geq 1/2$ for any λ. Therefore, for any $\varepsilon > 0$,

$$H(Q_\lambda) < H(Q_{\lambda+\varepsilon}).$$

This completes the proof. □

An alternative interpretation. To conclude, let us provide the following complementary intuition. At $\lambda = 0$, $\theta(0) = 1$ and no entry of z_0 is set to zero. This clearly corresponds to the easiest available example, since the learning starts at $t = 0$ by considering all possible available visual information. When θ start decreasing to $\theta(\lambda T) \approx 0.99$, only 1% of z_0 is suppressed (on average) and still almost all the information of the original dataset $\mathcal{Z}_0$ is available for training the network. But, as λ grows, $\theta(\lambda T)$ decreases and a bigger number of entries are set to zero. This complicates the task, requiring an improved effort

from the model to capitalize from the reduced uncorrupted information which is available at that stage of the training process.

After all, this connection between Dropout and Curriculum Learning was possible thanks to our generalization through Def. 3. Consequently, the original Dropout [Hin+12; Sri+14] can be interpreted as considering the single specific value $\overline{\lambda}$ such that $\theta(\overline{\lambda}T) = \overline{\theta}$, being $\overline{\theta}$ the constant retain probability on [Hin+12; Sri+14]. This means that, as previously found for the adaptive regularization (see Section 6.2.1), the level of difficulty $W_{\overline{\lambda}}(z)$ of the training examples z is fixed in the original Dropout. This encounters the concrete risk of either oversimplifying or overcomplicating the learning, with detrimental effects on the model's generalization capability. Hence, the proposed method allows to setup a progressive curriculum $Q_\lambda(z)$, complicating the examples z in a smooth and adaptive manner, as opposed to [Hin+12; Sri+14], where such complication is fixed to equal the maximal one from the very beginning (Fig. 6.5).

To conclude, let us note that the aforementioned work [Ren+14] proposes a linear *increase* of the retain probability. According to equations (6.162-6.164) this implements what [Ben+09] calls an anti-curriculum: this is shown to perform slightly better or worse than the no-curriculum strategy [Ben+09] and always worse than any curriculum implementation. Our experiments confirm this finding.

6.2.3 Experiments

In this Section, we applied Curriculum Dropout to neural networks for image classification problems on different datasets, using Convolutional Neural Network (CNN) architectures and Multi-Layer Perceptrons (MLPs). In particular, we used two different CNN architectures: LeNet [LeC+89] and a deeper one (conv-maxpool-conv-maxpool-conv-maxpool-fc-fc-softmax), further called CNN-1 and CNN-2, respectively. In the following, we detail the datasets used and the network architectures adopted in each case.

MNIST - A dataset of grayscale images of handwritten digits (from 0 to 9), of resolution 28×28. Training and test sets contain 60.000 and 10.000 images, respectively. For this dataset, we used a three-layer MLP, with 2.000 units in each hidden layer, and CNN-1.

Double MNIST - This is a static version of [Sri+15], generated by superimposing two random images of two digits (either distinct or equal), in order to generate 64×64 images. The total amount of images are 70.000, with 55 total classes (10 unique digits classes + $\binom{10}{2} = 45$ unsorted couples of digits) . Training and test sets contain 60.000 and 10.000 images, respectively. Training set's images were generated using MNIST training images, and test set's images were generated using MNIST test images. We used CNN-2.

SVHN [Net+11] - Real world RGB images of street view house numbering. We used the cropped 32×32 images representing a single digit (from 0 to 9). We exploited a subset of the dataset, consisting in 6.000 images for training and 1.000 images for testing, randomly selected. We used CNN-2 also in this case.

CIFAR-10 and **CIFAR-100** [KH09] - These datasets collect 32×32 tiny RGB natural images, reporting 6000 and 600 elements per each of the 10 or 100

classes, respectively. In both datasets, training and test sets contain 50.000 and 10.000 images, respectively. We used CNN-1 for both datasets.

Caltech-101 [FF+04] - 300 $\times$ 200 resolution RGB images of 101 classes. For each of them, a variable size of instances is available: from 30 to 800. To have a balanced dataset, we used 20 and 10 images per class for training and testing, respectively. Images were reshaped to 128×128 pixels. We used CNN-2 again here.

Caltech-256 [Gri+07] - 31000 RGB images for 256 total classes. For each class, we used 50 and 20 images for training and testing, respectively. Images were reshaped to 128×128 pixels. We used CNN-2.

For training CNN-1, CNN-2 and MLP, we exploited a cross-entropy cost function with Adam optimizer [KB14] and a momentum term of 0.95, as suggested in [Sri+14]. We used mini-batches of 128 images and fixed the learning rate to be 10^{-4}.

We applied curriculum dropout using the function (6.140) where γ is picked using the heuristics (6.141) and $\overline{\theta}$ is fixed as follows. For both CNN-1 and CNN-2, the retain probability for the input layer was set to $\overline{\theta}_{\text{input}} = 0.9$, selecting $\overline{\theta}_{\text{conv}} = 0.75$ and $\overline{\theta}_{\text{fc}} = 0.5$ for convolutional and fully connected layers, respectively. For the MLP, $\overline{\theta}_{\text{input}} = 0.8$ and $\overline{\theta}_{\text{hidden}} = 0.5$. In all cases, we adopted the recommended values [Sri+14, Section A.4].

Before reporting our results, let us emphasize that our aim is to improve the standard dropout framework [Hin+12; Sri+14], not to compete for the state-of-the art performance in image classification tasks. For this reason, we did not use engineering tricks such as data augmentation or any particular preprocessing, and neither we tried more complex (or deeper) network architectures.

In Fig. 6.8, we qualitatively compared Curriculum Dropout (green) versus the original Dropout [Hin+12; Sri+14] (blue), anti-Curriculum Dropout (red) and an unregularized, *i.e.* no Dropout, training of a network (black). Since CNN-1, CNN-2 and MLP are trained from scratch, in order to ensure a more robust experimental evaluation, we have repeated the weight optimization 10 times for all the cases. Hence, in Fig. 6.8, we report the mean accuracy value curves, representing with shadows the standard deviation errors.

Additionally, we report in Table 6.1 the percentage accuracy improvements of Dropout [Hin+12; Sri+14], anti-Curriculum Dropout [Ren+14] and Curriculum Dropout (proposed) versus a baseline network where no neuron is suppressed. To do that, we selected the average of the 10 highest mean accuracies obtained by each paradigm during each trial; then we averaged them over the 10 runs. We accommodated the metric of [TE11] to measure the boost in accuracy over [Hin+12; Sri+14]. Also, we reproduced for two datasets the cases of fixed layer size n or fixed $n\overline{\theta}$ as in [Sri+14, Section 7.3]. Here the network layers' size n is preliminary increased by a factor $1/\overline{\theta}$, since on average a fraction $\overline{\theta}$ of the units is dropped out. However, we notice that those bigger architectures tend to overfit the data.

Discussion. The proposed Curriculum Dropout, implemented through the scheduling function (6.140), improves the generalization performance of [Hin+12;

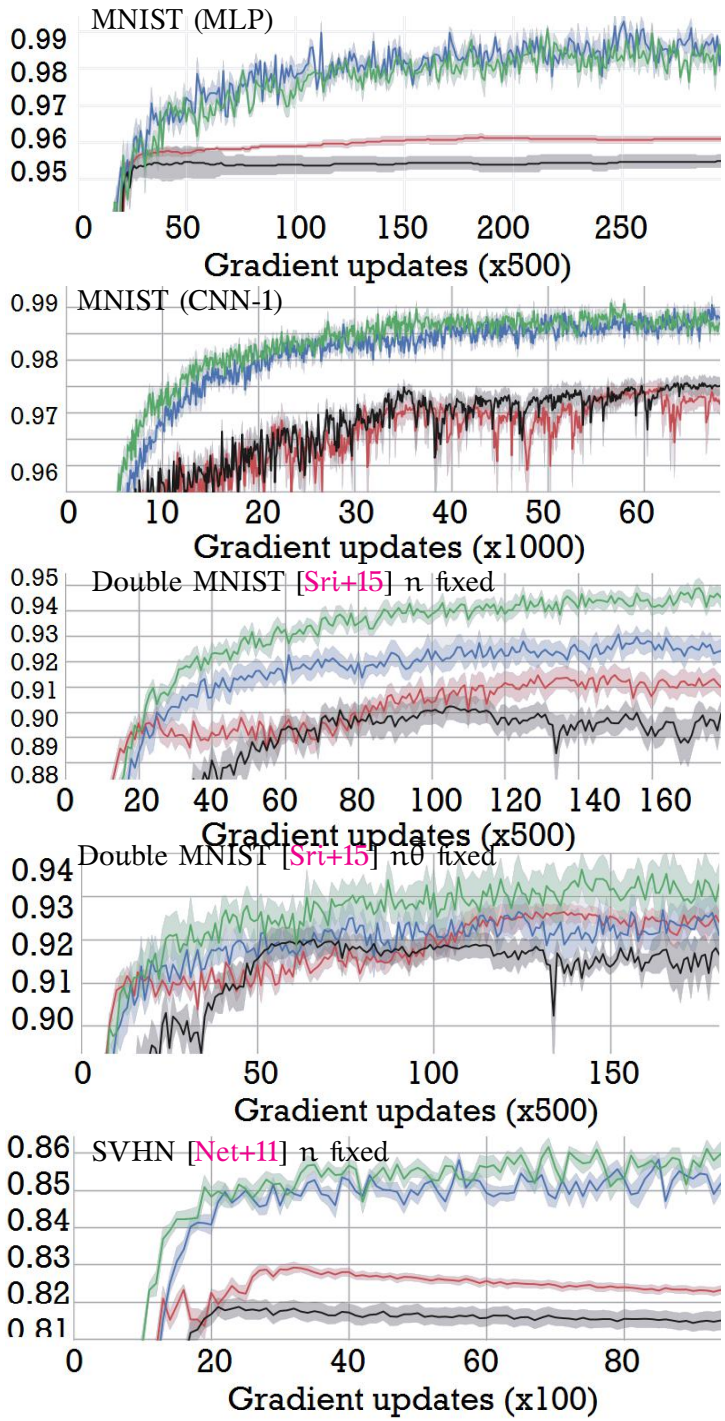


FIGURE 6.7: Curriculum Dropout (green) compared with regular Dropout [Hin+12; Sri+14] (blue), anti-Curriculum (red) and a regular training of a network with no units suppression (black). For all cases, we plot mean test accuracy (averaged over 10 different re-trainings) as a function of gradient updates. Shadows represent standard deviation errors. Best viewed in colors.

Sri+14] in almost all cases (*e.g.* , Caltech 256 [Gri+07], +0.87%, or Double MNIST n fixed, +0.93%). As the only exception, in MNIST with MLP, the scheduling is just equivalent to the original dropout framework [Hin+12; Sri+14]. Our guess is that the simpler the learning task, the less effective Curriculum Learning. After all, for a task which is relatively easy itself, there is

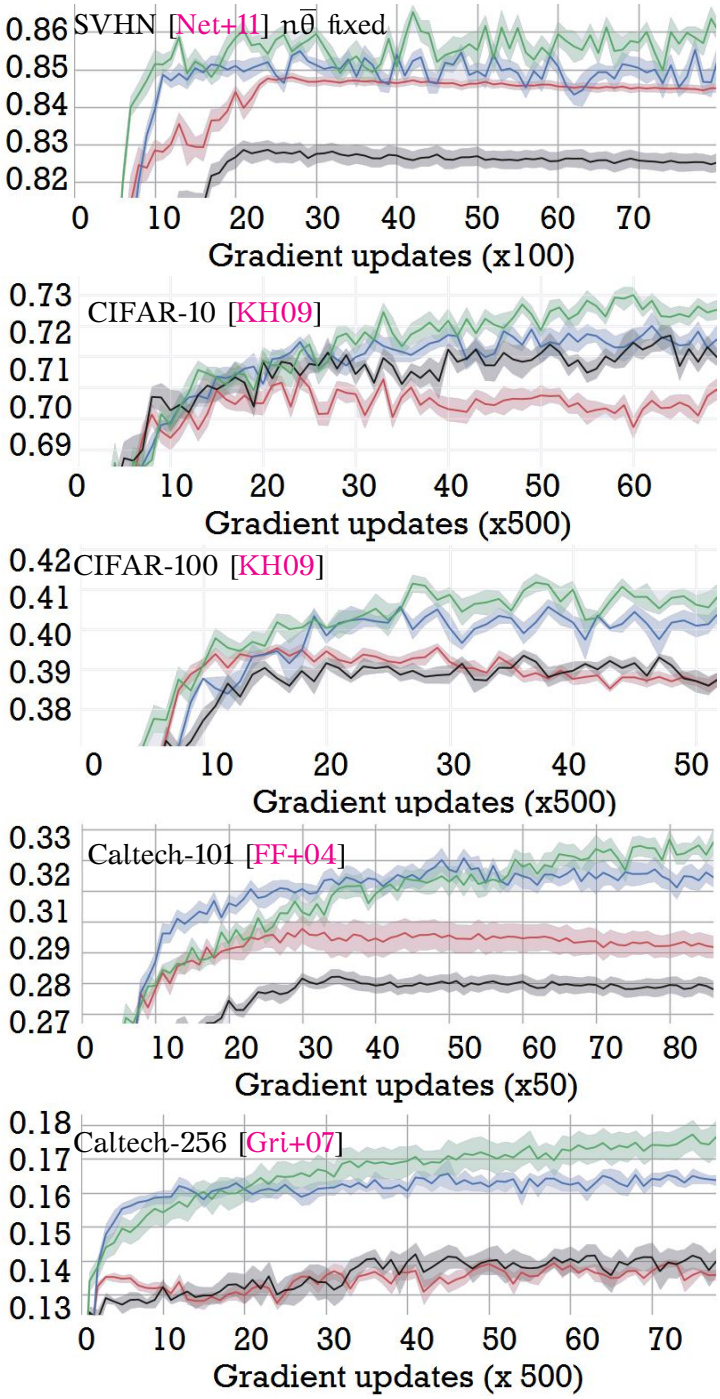


FIGURE 6.8: Curriculum Dropout (green) compared with regular Dropout [Hin+12; Sri+14] (blue), anti-Curriculum (red) and a regular training of a network with no units suppression (black). For all cases, we plot mean test accuracy (averaged over 10 different re-trainings) as a function of gradient updates. Shadows represent standard deviation errors. See also Figure 6.7. Best viewed in colors.

less need for "starting easy". This is in any case done with at additional cost nor training time requirements.

As expected, anti-Curriculum was improved by a more significant gap by our

Dataset	Architecture	Configuration (n or $n\bar{\theta}$ fixed)	Classes	Unregularized network	Dropout [Hin+12; Sri+14]	Anti-Curriculum	*Curriculum Dropout* (percent boost [TE11] over Dropout [Hin+12; Sri+14])
MNIST	MLP	n	10	98.67	**+0.38**	+0.04	*+0.36* (-5.3%)
	CNN-1	n		99.25	+0.15	-0.05	***+0.18 (20.0%)***
Double MNIST	CNN-2	n	55	92.48	+1.42	+0.73	***+2.35 (65.5%)***
	CNN-2	$n\bar{\theta}$			+0.87	+0.53	***+1.11 (27.6%)***
SVHN [Net+11]	CNN-2	n	10	84.63	+2.35	+1.17	***+2.65 (12.8%)***
	CNN-2	$n\bar{\theta}$			+1.59	+1.51	***+2.06 (29.6%)***
CIFAR-10 [KH09]	CNN-1	n	10	73.06	+0.22	-0.68	***+0.62 (182%)***
CIFAR-100 [KH09]	CNN-1	n	100	39.70	+1.01	+0.01	***+1.66 (64.4%)***
Caltech-101 [FF+04]	CNN-2	n	101	28.56	+4.21	+1.57	***+4.72 (12.1%)***
Caltech-256 [Gri+07]	CNN-2	n	256	14.39	+2.36	-0.22	***+3.23(36.9%)***

TABLE 6.1: Comparison of the proposed scheduling versus [Hin+12; Sri+14] in terms of percentage accuracy improvement.

scheduling: +1.65% on CIFAR-100 [KH09]. Also, sometimes it even performs worse than a non-regularized network (*e.g.* , Caltech 256 [Gri+07]). This is coherent with the findings of [Ben+09] and with our discussion in Section 6.2.2 concerning Annealed Dropout [Ren+14], of which anti-Curriculum represents a generalization. In addition, while neither regular nor Curriculum Dropout ever need early stopping, anti-Curriculum often does.

6.3 Conclusions

Within our analysis of dropout for matrix factorization, we present a theoretical analysis which, differently from previous works in the literature, is not affected by any sort of approximation. In the case of a fixed size of the factors d, we proved that the expectation computed over $r_1, \dots, r_d \sim \text{Bernoulli}(\theta)$ casts dropout for MF (6.3) into the fully deterministic optimization problem (6.1) where $\Omega = \Omega_{\text{dropout}}$. For any fixed d, the two problems are equivalent in a very strong manner since, for any $\mathbf{U}$ and $\mathbf{V}$, the two objective functionals are point-wise equal and, consequently, by either solving (6.3) or (6.1), the optimal solution $\mathbf{U}^{\text{opt}}$ and $\mathbf{V}^{\text{opt}}$ is the same.

Additionally we also showed a strong connection between nuclear norm regularization and dropout regularization. In particular, we began by noting the close similarity between Ω_{dropout} and the variation form of the nuclear norm, but then we demonstrated that with a fixed choice of θ the resulting problem allows the size of factorization to grow unbounded.

We also investigated the case of a factorization with variable size. When d varies, the regularizer Ω_{dropout} is pathologically promoting over-sized factorizations when θ is fixed. This motivated us in proposing an adapted choice for θ which, as defined in (6.73), depends upon the size of the factorization d and

the hyper-parameter p. This stage ensures that, not only the aforementioned problem is solved, but at the same time, we are able to guarantee that $\theta = \theta(d)$ as in (6.73) prevents other issues to arise. This is true because we demonstrate that the lower convex bound of $\frac{1-\theta(d)}{\theta(d)}\Omega_{\text{dropout}}$ is the nuclear norm squared. Ancillary, we took advantage of this result to prove that, the optimal dropout for MF factors immediately get for free the global optimum of the convex optimization problem (6.87). Since the latter is a convex (squared) nuclear norm regularization that, as we argumented, can be framed as an adaptive PCA that, also, learns from data the optimal size d (6.126) that should be used to reduce the dimensionality of the data.

Additionally, our results show a novel interpretation of dropout that suggests it enforces spectral sparsity and thus acts to promote low-rank solutions.

Finally, we have verified our theoretical predictions via experiments on both simulated and real data, and our results suggest a novel approach to linear subspace learning which is worthy of further study in various applications for artificial intelligence.

Afterwards, inspired by the previous theoretical findings, we have propose a scheduling for dropout training applied to deep neural networks. By softly increasing the amount of units to be suppressed layer-wise, we achieve an adaptive regularization and provide a better smooth initialization for weight optimization. This allows us to implement a mathematically sound curriculum [Ben+09] and justifies the proposed generalization of [Hin+12; Sri+14].

Through a broad experimental evaluation on 7 image classification tasks, the proposed Curriculum Dropout have proved to be more effective than both the original Dropout [Hin+12; Sri+14] and the Annealed [Ren+14], the latter being an example of anti-Curriculum [Ben+09] and therefore achieving an inferior performance to our more disciplined approach in ease dropout training. Globally, we always outperform the original Dropout [Hin+12; Sri+14] using various architectures, and we improve the idea of [Ren+14] by margin.

Chapter 7

Conclusions

In this thesis, we throughly investigated the framework of learning by correlation, a tool which is directly inspired by human cognition. That is, rather that inspecting absolute values of feature representations conveyed by the data, we tried to exploit the strength of their mutual correlation in order to boost the learning stage.

In fact, for the sake of human action recognition from skeletal data, each position of each single joint in the skeleton is not really important per se, but, instead, it must be compared with the remaining joints in order to proficiently track human poses in time. For instance, the task of disambiguating `high-five` and `punching` activities is arguably hard if one is allowed to only consider the position of the elbow are not really evocative (since always describing by a forward trajectory). Rather, a global description is needed in order to inspect whether the whole arm is describing a rising trajectory (`high-five`) or a forward one (`punching`). In this thesis, as a principled mathematical tool to achieve such description, we applied a spatio-temporal covariance representation which is capable of capturing all possible binary correlations between joints in the human skeleton. As explained in Chapter 3, such class of approaches has emerged as a state-of-the-art approach. Yet, it suffers from the following issues.

- By means of classical covariance operators, linear mutual relationship within the data can be captured only.
- When looking for a max margin classification of covariance operators on the manifold to which they belong, kernelized classifier needs to be used and, despite the sound performance, they are affected by a scalability issue.
- When capturing mutual correlations of skeletal joints, all pairwise mutual relationships between joints are captured, but, it can be argued that only some of those are really relevant for discriminating actions, being the remaining ones not informative or, worse, misleading.

For each of the previously highlighted problems, in Chapter 3, we proposed the following ad-hoc solutions.

- If non-linear correlations are postulated to be relevant, one may think to capture them by preliminarily transforming the data with a feature map and then capturing linear mutual relationships in the transformed space. A drawback arise from this approach, feature transformation needs to be explicitly computed: such transformations are always very complicated. In addition kernelized classifier has proved to be often superior to linearized ones when using kernels which are associated to infinite dimensional feature maps. Unfortunately, explicitly using infinite dimensional feature maps for the computation of covariance representation is impossible. In our work, we propose to exploit infinite dimensional feature representation for covariance estimation by implicitly computing them via a kernel: in other words, we recover the kernel trick for covariance representation. With a sound mathematical analysis, we provided a theoretical foundation for our approach which simply allowed us to compute a more powerful covariance representation at the same computational cost of the previous one. Experimental evidences suggested that our approach is superior to existing state-of-the-art covariance based methods for 3D human action recognition of public benchmark datasets.

- Mathematically covariance representations are symmetric and positive-definite matrices and, in order to classify them proficiently, max margin approaches based on geodesic distance (computed with kernels) have been proved to be powerful. However, computing geodesic distances for all pairs of input covariance representations compromise scalability towards the big data regime. In this thesis we proposed a remedy for such burden, by proposing a novel kernel approximation in order to devise a compact feature representation which, combined with a linear classifier, is able to implicitly implement the exact kernel machine with geodesic distance buy in a scalable manner since we replace the evaluation of a distance similarity (quadratic complexity as a function of the data size) with a computation of a vectorial embedding (linear complexity). Our approximation writes as a random feature map which is an unbiased estimator of the Log-Euclidean kernel and, on top of that, the variance of the proposed estimator can be bounded with an inversely cubic function with respect to the data dimensionality. If comparing with alternative approximated schemes, our approach allows for a superior performance achieved with a more compact representation.

- We exploit a neural-network approach based on back-propagation so that we let the data decide which correlations among skeletal joints are more worth to be trusted for an effective human action recognition. This is implemented as a shallow architecture in which, after the computation of covariance representation, a weighted fully connected layer is adopted as to re-modulated each temporal correlation patterns in order to re-scale its relevance within the learning stage. Due to she shallowness of the method, training is fast even on CPU and, even if comparing with much deeper neural network, our proposed approach ensures a favorable performance. Such capability of pairing a solid performance with model compactness allows us to empirically validate the claim that as soon as

the actions' kinematics is efficiently modeled (in this case, with a spatio-temporal covariance representation), there is no need for the architecture to be deep nor recurrent in order to accomplish action recognition in an effective manner.

Leveraging on our findings, in Chapters 4 and 5, we complicated the complexity of the type of correlations we captured. Indeed, after measuring the degree of correlation exhibited by different components of one feature representations with which data are encoded, we considered the following two generalizations.

1. We assume to encode the same raw data with different feature representation in parallel that we called *views*. Multimodal representations are surely effective in capturing complementary aspects of the recognition task, but, as a drawback of such approach, how to combine those representations at the level of model prediction is not an important task also under the perspective of achieving a robust prediction. In fact, when data annotations are not precise due to human or systematic errors, relying on multiple views to represent the data is a viable tool to spot which annotation are outliers which can mislead the learning stage of the model.

2. We tackle the problem of transfer learning in which we train one model on a fully labeled dataset (*source domain*) and we are interested in applying it to a novel dataset - the *target domain* - on which no annotations are available. Since the dataset are different, as they are the two data distributions and, consequently, directly applying to target the model trained on the source leads to a performance degradation. In order to avoid this problem, we use correlation as a tool to bridge the semantic gap and prevent such degradation by aligning the source and data distributions.

In details, the previous problems have been addressed by the following two proposed paradigms

1. In Chapter 4, we propose a robust regression to exploit the Huber loss for the sake of learning how to combine multiple views in a unique holistic pipeline applied to scalar regression problems. In fact, we observe that, due to the human engagement in providing data annotations, some of those can be mistaken. Also, due the low-resolution data as the ones processed in video-surveillance applications, some types of annotations, such as providing the ground truth number of pedestrians in a given scene, may be extremely difficult to produce. Inspired by all this consideration we exploit the richness derived from multiple parallel encodings (here called *views*) to encode the data: by measuring to which extent annotations correlate to all those multiple representations, we are able to automatically spot outliers in the label. Such component is embedded within a manifold regularized multi-view scalar regression framework which automatically balances the amount of supervision by removing those annotations which are detected as corrupted and by exploiting the corresponding input data in a fully unsupervised way. Despite the complex framework considered, we are able to carry out optimization through a refinement scheme that iterates between two tasks: 1) learning from data the sensitivity threshold ξ which quantifies the maximum level of reconstruction error tolerated within the training set and 2) for

that learnt ξ optimize the Huber loss H_ξ with a closed-form solution. In comparison with existing algorithms proposed in the literature to optimized the Huber loss, we are not asked to heuristically fix ξ in H_ξ nor we need to opt for approximated solution. Moreover, in practical terms, we certify the solidity of our proposed frameworks by means of a broad experimental validation in which we outperformed with one single techniques classical approaches for regression on UCI benchmarks, customized methods for learning from noisy labels in binary classification tasks and state-of-the-art methods for crowd counting with hand-crafted descriptors.

2. Afterwards, we push correlation towards the dimension of capturing mutual correlations among two different domains, which share the same visual categories but not the same exact data distribution. In such a condition, a model can not be transfered across domains without encountering a performance degradation: in order to tackle this issue, we take advantage of the well established paradigm of correlation alignment in order to bridge the semantic gap between dataset, attenuate the domain shift and ultimately reduce the phenomenon of performance degradation. In spite of success of correlation alignment, two problems related with this approach have been not addressed so far.
   First, since correlation can be formalized as a symmetric and positive definite (SPD) matrix, deep correlation alignment techniques which attempt to match correlations through Euclidean distance are arguably suboptimal if compared with methods which instead accomplish alignment directly on the SPD manifold by means of geodesic distance.
   Second, when augmenting classical supervised training losses with correlation alignment penalty, the impact of the latter on the former is usually controlled by means of a Lagrangian multiplier. In order to fine-tune the latter, devising a cross-validation scheme is actually problematic in unsupervised domain adaptation: cross-validating on a sub-portion of the source domain is likely to be not informative about the performance on the target domain due to the domain shift. Also, direct cross-validation on the target domain is unfeasible due to the operative condition in which the target domain is assumed to be totally unlabelled.
   In Chapter 5 we solve all those problems by proposing MECA - Minimal-Entropy Correlation Alignment - in which, in addition to carry out a geodesic alignment between correlations with the Log-Euclidean metric, we also found out that target entropy is a reliable criterion to cross-validate the Lagrangian multiplier. In fact, on the one hand, we are able to effectively find the value of the Lagrangian multiplier which results in the top performance on the target and, on the other hand, we do so *without* exploiting any label in the target domain. This is very original with respect to classical cross-validation strategies which are (partially) supervised: differently, our target entropy criterion is actually fully unsupervised. Those favorable theoretical properties translate into a superior performance when compared to deep learning methods for unsupervised domain adaptation applied to digit classification and cross-modal object categorization.

As the last point of our analysis related to correlation, after having widely

explored techniques to take advantage of correlation as a cue to boost learning, we also consider the case where correlation actually acts as a sort of noise which need to be removed in order to not damage the learning stage. In fact, we considered the situation where either the raw data or the deployed feature representation is affected by excessive correlations among variables which ultimately result in over-redundancies, the latter being detrimental for the generalization capabilities of the model that needs to be learned. Eventually, such issue is much more urgent in the case of modern paradigms of end-to-end learning of data driven representations and, in order to remove excessive correlations among units in the network, dropout has established as an effective countermeasure against overfitting. That is, dropout seems to act as a regularizer: in spite of that it is formally very different with respect to usual (additive Tikhonov) regularization strategies that are very common within the machine learning community. In Chapter 6, we shed light on the connection between dropout training for matrix factorization models and a peculiar type of additive regularization with the nuclear norm, which is a well used tool to promote spectral sparsity. The key to achieve such connection is to allow the dropout rate to be variable with respect to the size of the factorization: in such a case, we formally demonstrate that dropout is acting on this peculiar model as nothing but PCA.
Inspired by the theoretical findings we elicit in the case of matrix factorization, we apply the same idea of a variable dropout rate to the case of deep neural network training. That is, we postulate that, in usual schemes of training networks by means of several epoch of back-propagation, over-fitting is likely to occur only within the last epochs. Therefore, still applying dropout to early stage of the learning is arguably suboptimal since regularization is simply not necessary at the beginning of the training. We efficiently implemented this idea through a temporal scheduling on the dropout rate by means of a negative exponential function which smoothly decreases the amount of units that are kept at each layer while back-propagating errors. In formal terms, we demonstrate that our approach can be interpreted as a peculiar form of curriculum dropout [Ben+09], that is, we apply a "starting small" approach in which easier examples (that is, not corrupted by Bernoulli noise) are showed to the network for training before than harder ones (that is, the corrupted ones). The originality with respect to Curriculum Learning is that, differently from it, our method does not require to explicitly partition the data into easier and hard sub-parts: antithetically, the starting small approach is implicitly done within the network by just smoothly modulating the amount of Bernoulli random noise exploited at the level of either input data or intermediate representations. Experimental evidences suggest that the proposed variations of dropout training is never inferior to classical dropout training, often yielding to improved generalization capabilities of the models quantified in a superior performance on action recognition benchmarks.

Future Works. Despite the many directions of analysis investigated in this thesis, some future works can be sketched. After applying spatio-temporal covariance representations to the problem of action recognition, we empirically demonstrated the effectiveness of capturing second order statistics. Yet, one could argue that higher order statistics maybe more effective in capturing finer nuances of the data and this is surely and interesting direction to be explored.

In the context of learning by correlating multiple modalities applied for data

representation, we will be interested in generalizing the usage of the Huber loss to the case of a vector-valued regression, possibly deriving an efficient optimization strategy in a similarly theoretical semi-supervised multi-view learning paradigm.

Leveraging the formal analysis of dropout for matrix factorization, an interesting direction will be to apply a similar theoretical investigation in order to cover a deeper model. This would push the analysis more closed to the deep neural network for which dropout has been originally designed and we could explain what is the factual regularization scheme that is induced by dropout training even in the case of currently adopted deep neural networks.

In the case of capturing mutual correlation in the data for the sake of unsupervised domain adaptation, the following direction could be of potential interest. In our approach, we globally aligned the source and target domain by checking second-order statistics. As another alternative direction to do so, the recent success of cycle consistency approaches in domain adaptation [Hae+17c; Hae+17b] has proved the effectiveness of adopting a more local approach in which source and target are traversed by means of interpolating trajectories across data points, requiring those trajectories to be a closed inside the same class. At a first glance, such global and local approaches for domain adaptation appear as complementary strategy. Therefore, another possible future direction will be to deploy a theoretical and empirical analysis to certify to which extent those two classes of methods are compatible in aligning source and target distributions.

, we could investigate whether such approach of traversing the data manifold by interpolating across similar classes in the source and target domain can be combined with the geodesic alignment of second order statistics. More generally, we could eventually test whether the proposed geodesic alignment of correlation statistics is compatible with alternative domain adaptation methods.

Appendix A

Intention from Motion: Correlating Action's Kinematics and Overarching Intent

Abstract

This Appendix aims at investigating the action prediction problem from a pure kinematic perspective. Specifically, we address the problem of recognizing future actions, indeed human intentions, underlying a same initial (and apparently unrelated) motor act. This study is inspired by neuroscientific findings asserting that motor acts at the very onset are embedding information about the intention with which are performed, even when different intentions originate from a same class of movements. To demonstrate this claim in computational and empirical terms, we designed an *ad hoc* experiment and built a new 3D and 2D dataset where, in both training and testing, we analyze a same class of grasping movements underlying different intentions. We set a broad baseline of state-of-the-art action recognition and prediction pipelines and we show that each grasping conveys enough information to allow a reliable prediction of the human intention. Afterwards, we investigate how much those discriminants generalize across subjects, discovering that each subject tends to affect the prediction by his/her own bias. As a first attempt to cope with such issue, we propose a two-stage pipeline where subjects' identification is used a preliminary step before carrying out action recognition: this allows to take advantage of action classifier which are not generically trained on a group on agents, being instead directly *personalized* on the subject whose actions need to be recognized. Second, inspired by domain adaptation, we propose to interpret each subject as a domain, leading to a novel subject adversarial paradigm to *de-personalize* intention prediction, ultimately, bridging the shift related to training a system on different subjects with respect to the one adopted in testing. Experimentally, we demonstrate the both the personalization and the de-personalization frameworks favorably cope with our new problem and with action recognition benchmarks as well.

A.1 Introduction

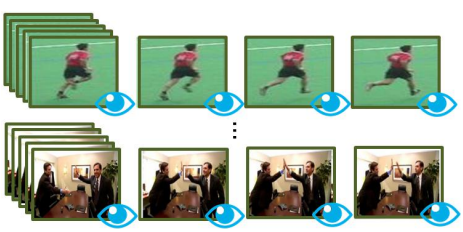
Action and activity recognition are surely intriguing and most active areas in computer vision. The task here typically consists in the classification of a *fully observed* action or activity. More recently, the community has also started to investigate a variant, extending the paradigm to the "early" activity recognition, which aims at recognizing an action before it is fully disclosed. Early activity recognition is sometimes improperly confused with action prediction, improperly because such works are not really predicting an action, but rather they are actually classifying an action from its beginning, *i.e.* , they identify an action from the observation of the onset of that *same* action. The actual action prediction problem consists instead in the classification of future actions considering all the events occurring up to a certain instant [CRC14]. As a different paradigm, we aim here at introducing a brand new challenging action prediction problem consisting in the prediction of human *intentions*, defined as the overarching goal embedded in an action sequence.

In Fig. A.1, we show the different paradigms for action and activity analysis and our proposed new concept for intention prediction. The novelty stands from the fact that intentions cannot easily be predicted using discriminant previous information extracted from a certain anticipative data stream since, unlike the other paradigms, such data displays the same class of motor act which can be performed with different intentions. Despite this, in general, the prediction of intentions still remains a manageable, yet rather complex problem, as we will show in the following.

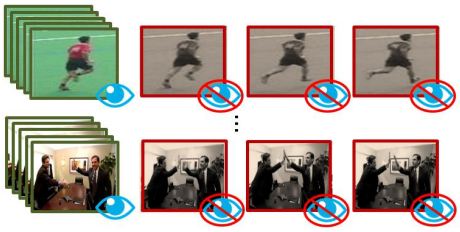
Previous attempts in classifying future or unfinished actions utilize developing motion patterns which are specific of the subsequent actions, since they contain some cues that undoubtedly help the recognition. For instance, if the goal is understanding whether two people are going to shake their hands or to give a high-five, by just looking at the first part of their interaction, a low wrist height can be an evidence of a handshaking [Von+16; Lan+14]. Further, another important aspect of the entire activity recognition problem is that the current techniques are mainly exploiting the *scene context* to support the classification ([Cao+13; FZ14; CRC14; Xie+13; Wal+14; Min+11; Kil11; Elk14] and [Bub+13]), *i.e.*, the objects present in the scene and the knowledge about the actions associated to them are cues that can be utilized to help in making a correct inference of the ongoing action to be recognized. However, although this information could help, it can be insufficient to solve the task or, worse, the context may not always be available or easily recognizable, being also misleading when the scene is too noisy or cluttered [ZB15].

In any case, an important source of information to disambiguate intentions can be provided by the kinematics of the movement [Sta+12].

In fact, recent findings from behavioural neuroscience indicate that how a motor act is performed (*e.g.*, grasping an object) is not solely determined by biomechanical constraints imposed by the object's extrinsic and intrinsic properties with which one is interacting but it depends on the agent's intention (*e.g.*, to pass vs. to use the object [Ans+14; Ans+15]). Since the same cerebral areas are used in both motor planning and intent understanding [Ozt+05], this

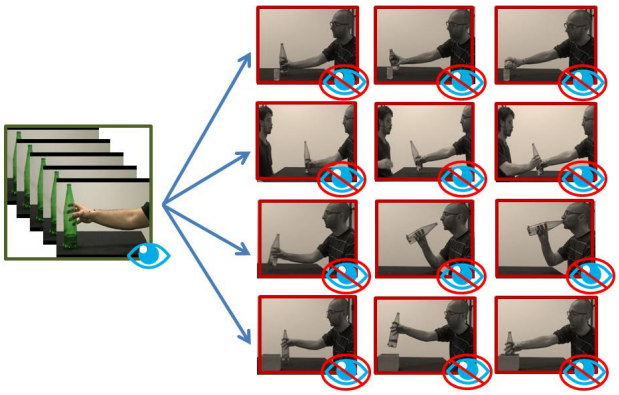


(a) Action/activity recognition



(b) Early activity recognition

(c) Action prediction



(d) The proposed paradigm of intention prediction

FIGURE A.1: Four different paradigms. (a) Action/activity recognition: the full sequences is exploited for classification ("running" for the top sequence up to "high-five" for the bottom). (b) Early activity recognition: a few initial frames are observed and classification rely upon such incomplete information. (c) Action prediction: future actions are predicted on the basis of all past events which are class-specific. For instance, in the top sequence a standing up activity leads to predict a "kissing", while, in the bottom, a conversation between a group of friends anticipates a "high-five". (d) Intention prediction: the same class of motor act (in the picture, grasping) is analyzed to explain why the motor act itself has been displayed, predicting its underlying intention (in the picture, from top to bottom, Pouring, Passing, Drinking or Placing).

suggests the fascinating possibility that predicting intentions is viable *from kinematics only*.

To this end, we propose a new dataset as a test-bed to investigate the feasibility of inferring intention from motion so as to provide a proof of concept for this task. A set of experiments was designed in which subjects were asked to grasp a bottle, in order to either 1) pour some water into a glass, 2) pass the bottle to a co-experimenter, 3) drink from it, or 4) place the bottle into a box. The dataset is composed by a) 3D trajectories of 20 motion capture (VICON) markers outfitted over the hand of the participants and b) optical video sequences lasting about one second, with an occlusive camera view in which only the arm and the bottle are visible. The goal is to classify the intentions associated with the observed grasping-a-bottle movement, *i.e.* to predict the agent's intention.

Even if the literature presents some action prediction methods [Lan+14; Ryo11; Von+16; HD12; Wal+14], the experiments therein included are typically performed on standard action recognition datasets, just adapted to the new task, and often considering the start of the same action, which of course facilitates the prediction.

On the contrary, our new dataset is explicitly designed for intention prediction, a problem never considered before in computer vision in the terms we posit. It is also important to note that, despite the apparent simplicity of the experiments, this scenario constitutes an actual proof of concept which can be further extended to actual applications. Indeed, an object, namely a knife, can be grabbed with very different intentions, *e.g.*, cropping an apple or attacking a person. In more into-the-wild applications, it would be extremely valuable to infer whether a subject who is standing in front of a bank counter and grabbing something from the pocket, will pick his wallet and deposit money or, instead, will extract a gun and attempt a robbery.

Further, in crowd behaviour analysis, it would be paramount to detect whether the apparently casual motion patterns of an individual/group of people can forerun a fight or, in social robotics, to provide a robot of the capability to read human intentions in order to figure out her action, hence providing to the subject the feeling of a more realistic engagement. Evidently, in all these cases, the discrimination must rely on the kinematics exclusively, being the context not informative.

Another aspect which should be considered in the design of methods coping with action recognition problems and related variants is the capacity of *generalization*. This results a crucial point for intention prediction as well. Specifically, since the same class of anticipative motor acts subsuming different intentions is executed by several subjects, not only we have to figure out intention-specific discriminants from similar motor acts, but such discriminants should be also transversal (*i.e.* , invariant) across different subjects. In fact, realizing that a certain bias is associated to the subjects executing the grasping actions, we tried to exploit such information to our advantage to increase the generalization capability of our method. In order to cope with this additional complexity in an effective way, we propose a novel approach derived from the domain adaptation research which considers each subject as a domain and adopt a subject-adversarial training pipeline to generalize better

among the subjects. This approach showed the best performance in our test case, and also promising results in classic action recognition frameworks are obtained.

To sum up, this work is characterized by the following main contributions.

(a) We introduce the new problem of Intention from Motion. That is, from the same observable "neutral" motor act - used in both training and test phases - we classify the underlying intention using solely the motion information, without using any contextual cue.

(b) We propose a 3D & 2D dataset, specifically aimed at the prediction of human intentions. This dataset is designed in a principled way by defining four intentions (Pouring, Passing, Drinking, Placing) performed by independent naive subjects, which are all forerun from the very similar initial grasping-a-bottle movement, while avoiding bias which can affect the subsequent performance analysis. To the best of our knowledge, this is the first time a dataset has been explicitly designed for intention prediction. The dataset is publicly available for research purposes [1].

(c) We set a broad baseline analysis of existing 3D, 2D and multi-modal (3D+2D) techniques, as well as experimenting existing action prediction pipelines on our dataset: in all cases, we register a performance which exceeds the random chance level, clearly certifying that such general problem is affordable.

(d) Once discovered the specific cues which discriminate each single intention, we solve the classification task by devising an original prediction pipeline which greatly benefits from the automatic recognition of the subject's identity in order to boost the accuracy in anticipating his/her intention.

(e) We propose a novel method which explicitly addresses the biases associated to the human subjects performing the initial grasping action, an issue particularly affecting the intention prediction problem, even more severely than the other action recognition paradigms. In particular, we discovered an inherent inter-subject variability and intra-subject similarity of the motor acts when performed by different and same subject(s), respectively, and we devised a method aimed at exploiting such information to improve its generalization ability, which is derived from the domain adaptation (DA) research. This is done by interpreting each training subject as a source domain and the unknown testing subject as target domain, being testing intention labels *never* used in training.

   This method is named Subject-Adversarial Domain Adaptation (SADA) and it is formulated as a standard *unsupervised domain adaptation* problem [Gan+16] where unannotated testing trials are used to promote both intention discrimination and subjects' confusion. As a generalization of SADA, we also consider the case where the testing trials are never exploited at all: the adaptation is in this case performed in a complete blind manner between all the training subjects only (*i.e.* , trials of the testing subject are not processed by the system in any way during training). This latter method, called Blind-SADA, can be interpreted as a generalization

[1] https://www.iit.it/it/datasets/intention-from-motion-dataset.

of the unsupervised domain adaptation setting where, differently from typical frameworks (*e.g.* , [DM06]), the target data is not only unlabeled, but totally unknown.

Experimental evidences support the effectiveness of our approach for intention prediction and also for standard action recognition on benchmark datasets.

The rest of this Appendix is structured as follows. In Section A.2, we report some previous works from the early activity recognition and prediction literature. Section A.3 introduces our novel dataset which is intensively benchmarked throughout the 3D, 2D and multi-modal baseline analyses in Sections A.4, A.5 and A.6, respectively. In Section A.7, we propose our novel intention prediction technique after a deep investigation of our test-bed dataset. Finally, Section A.9 draws the conclusions and sketches future works.

A.2 Related Work

In this Section, we briefly report the most relevant works from the existing literature, which either deals with early activity recognition or action prediction.

Ryoo [Ryo11] devise a system to infer the ongoing activity by only analysing its *onset*, *i.e.* its beginning. This is done with a dynamic programming method to match an extension of classical bag-of-features representation which allows to capture the temporal correlation of descriptors. Hoai and De la Torre [HD12] design a max-margin event detectors to address the problem of the early recognition of a specific human emotion after it starts but before it ends. Yu et al. [Yu+12] propose a local approach to categorize actions from their beginning. The temporal-dependencies between different spatial location are implemented into a probabilistic graphical model fed by histogram features. Cao et al. [Cao+13] split a complete action into temporal segments which are further represented by means of sparse coding, so that actions are recognizable from incomplete data. Ryoo et al. [Ryo+15] tackle early activity recognition from egocentric videos: the task is detecting the so-called *onset signature*, a bunch of kinematic evidence which has strong predictive properties about the last part of the observed action. Some works have attempted to investigate how much of the whole action is necessary to perform a classification: Davis and Tyagi [DT06] adopt a generative probabilistic framework to deal with the uncertainty due to limited amount of data, while Schindler and Van Gool [SG08] try to answer the aforementioned question using a similarity measure between the statical and the motion information extracted from videos. Soran et al. [Sor+15] devise a notification system for daily activities where, for instance, the detection of an ongoing milk boiling alerts the human user. Xu et al. [Xu+15a] subdivide the beginning of a video into a bunch of snippets, and the final ending is predictable through a ranking model which simulates Internet query auto-completion. The early recognition is also tackled by Soomro et al. [Soo+16] by combining a conditional random field data representation with SVM for the prediction; Ma et al. [Ma+16] approach the same problem by combining LSTM and CNN architectures. Kong and Fu [KF16] cast SVM as an action prediction machine by building a composite kernel on top of a dense extraction of spatio-temporal features.

Li et al. [Li+12] use a random tree to model all the kinematics up to a certain instant, thus constraining the prediction of the most likely action (*e.g.*, predicting "grab an object" if "reach an object" is detected). Huang et al. [HK14] face activity forecasting for human interactions: the acts of an agent induce a cost topology over the space of reactive poses where the response of the co-agent can be retrieved. Lan and Savarese [Lan+14] develop the so-called *hierarchical movemes* to model human actions at multiple levels of granularities. Vondrick and Torralba [Von+16] uses a deep neural network trained over 600 hours of videos. During training the net exploits videos to learn to predict the representation of frames in the future and the last fully connected layer allows to perform classification over different future endings. Jain et al. [Jai+16] combine RNNs and spatio-temporal graphs to devise a structured temporal modelling pipeline which is applied to action prediction.

One common aspect of both (early) activity recognition and action prediction is that contextual information is frequently used to perform the classification. Indeed, once the objects present in a scene are detected, the object-object or object-person relationship can be modelled by several probabilistic architectures (*e.g.* , graphical models [CRC14; FZ14; LF14] or topic models [KS13; Min+11]). Among the works which directly model the context inside the algorithms, some of them deal with the prediction of future trajectories of moving objects (vehicles or pedestrian) [Kit+12; Wal+14; Yam+11; Xie+13] by estimating the spatial areas over which such objects will most likely pass with respect to those which are excluded by this passage (*e.g.* , car circulations over sidewalks [Wal+14]).

In this Appendix, unlike all the aforementioned works, we are not classifying actions from their very first beginning, but we aim at predicting *intention from motion*, a brand new challenge in action prediction consisting in recognizing (*i.e.* , anticipating) different intentions which finalize the same class of motor act, distilling from it the discriminative motion patterns characterizing the specific intention, while fully neglecting any contextual information.

A.3 The Dataset

Seventeen naive volunteers were seated beside a 110×100 cm table resting on it elbow, wrist and hand inside a fixed tape-marked starting point. A glass bottle was positioned on the table at a distance of about 46 cm and participants were asked to grasp it in order to perform one of the following 4 different intentions.

1. **Pouring** some water into a small glass (diameter 5 cm; height 8.5 cm) positioned on the left side of the bottle, at 25 cm from it.
2. **Passing** the bottle to a co-experimenter seating opposite the table.
3. **Drinking** some water from the bottle.
4. **Placing** the bottle in a cardboard $17 \times 17 \times 12.5$ box positioned on the same table, 25 cm distant.

After a preliminary session, in which participants are familiarized with the execution, each subject performed 20 trials per intention. The experimenter visually monitored each trial to ensure exact compliance of these requirements.

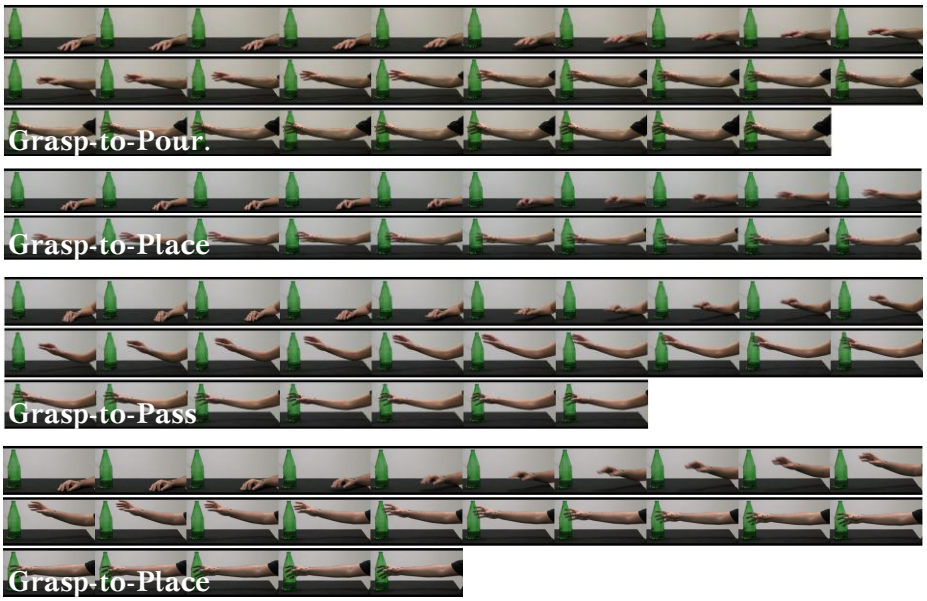


FIGURE A.2: The proposed problem of intention prediction. By only inspecting an apparently unrelated grasping-a-bottle motor act, we want to infer whether the latter is finalized to 1) pour some water into a glass, 2) pass the bottle, 3) drink from it or 4) place the bottle in a box (from top to bottom). We face this problem in pure kinematic terms: the context has been totally marginalized out.

FIGURE A.3: The disposition of the VICON marker on the subject's hand.

In order to homogenize the dataset, we completely removed trials judged imprecise. Thus, the final dataset includes 1098 trial (253 for pouring, 262 for passing, 300 for drinking and 283 for placing) and, for each of them, both 3D and video data have been collected. 3D marker trajectories and video sequences are acquired from the moment when the hand starts from a stable fixed position up to the reaching of the object, and both are exactly trimmed at the instant when the hand grasps the bottle, removing the following part. Our controlled setting constitutes an actual *worst-case scenario* since fixing the parameters for all the trials across subjects (*e.g.*, table size, position and size of the box, position of the co-experimenter, etc.), especially the starting hand position and the bottle location, we put ourself in *neutral* conditions, removing possible subjective biases which might affect the classification performance (*e.g.*, some intentions might be better discriminated if starting hand position would have been left free). Moreover, any other more complex activity forerunning an intention might provide further discriminant information for the prediction of the future action, with respect to a single, simple, arm movement.

3D kinematic data. Near-infrared 100 Hz VICON system was used to track the hand kinematics. Nine cameras were placed in the experimental room and each participant's right hand was outfitted with 20 lightweight retro-reflective hemispheric markers (see Figure A.3). After data collection, each trial was individually inspected for correct marker identification and then run through a low-pass Butterworth filter with a 6 Hz cutoff.

Globally, each trial is represented with a set of 3D points describing the trajectory covered by every single marker during execution phase. The x, y, z marker coordinates only consider the reach-to-grasp phase, the following movement is totally discarded. Indeed, the acquisition of each trial is automatically ruled by a thresholding of the wrist velocity $v(t)$ at time t, acquired by the corresponding marker. Being $\varepsilon = 20$ mm/s, at the first instant t_0 when $v(t_0) > \varepsilon$, the acquisition starts and it is stopped at time t_f, when the wrist velocity $v(t_f) < \varepsilon$.

2D video sequences. Movements were also filmed from a lateral viewpoint using a fixed digital video camera (Sony Handycam 3-D) placed at about 120 cm from hand start position. The view angle is directed perpendicularly to the agent's midline, in order to ensure that the hand and the bottle were fully visible from the beginning up to the end of the movement. It is worth noting that the video camera was positioned in a way that neither the box (Placing), nor the glass (Pouring), nor the co-experimenter (Passing) were filmed. Adobe Premiere Pro CS6 was used to edit the video in .mp4 format with disabled audio, 25 fps and 1280×800 pixel resolution. In order to format video sequences in an identical way to 3D data, each video clip was cut off at the exact moment when the bottle is grasped, discarding everything happening afterwards. To better understanding how demanding the task is, note that the actual acquired video sequences encoding the grasping last for about one fourth of the future action we want to predict: see Figure A.2. Consequently all the sequences result about 30 frames long.

Before proceeding, let us conclude with two additional remarks.

1. In all the experiments reported in this Appendix, either dealing with 3D or 2D data, we consider all the possible pairwise comparisons between intentions and the all-class one. We select *one-subject-out* testing procedure, that is, we compute seventeen accuracies, training our system on all the subjects except the one we are testing, then we averaged all the accuracies to get the final classification results.

2. If compared with existing action recognition datasets, the controlled experimental conditions of our dataset seems a limitation. For instance, MPII-CAD [Roh+12] and Salad 50 [SM13] cover more articulated (cooking) actions, while UCF-101 [Soo+12] and HDMB51 [Kue+11] collect YouTube videos, thus guaranteeing a broad variability of backgrounds and context. Conversely, we deliberately designed our case study in order to properly answer the question if the kinematics of the same ongoing action is enough informative to discover the intention which caused the following action. Indeed, the uncontrolled and real-world scenarios of the YouTube videos (such as in UCF-101 and HDMB51) may accidentally enrich the context with some cues which actually facilitate the prediction. Moreover, different future actions frequently begin with a quite different onset, *e.g.* , two persons *approach* each other before a "kissing" action occurs, or people *rise their hands* before a "high-five" action is carried out

3D results (%)	Kinematic Features			DTW		Covariance-based	
	F_{local}	F_{global}	F_K	K-nn	$\mathcal{L}$+SVM	H-COV	ker-COV
Pour vs. Place	79.70	86.10	84.32	87.86	83.28	90.23	**91.87**
Pour vs. Drink	72.15	70.36	76.48	67.06	83.59	91.43	**91.58**
Pour vs. Pass	76.55	67.39	**82.81**	66.49	81.98	80.43	81.69
Pass vs. Drink	63.10	68.05	70.75	54.29	82.53	87.30	**87.64**
Pass vs. Place	62.60	64.38	69.44	64.37	**82.27**	76.50	75.46
Drink vs. Place	64.40	71.41	73.72	71.51	90.74	89.63	**91.24**
All-class	45.08	48.01	55.13	40.86	63.10	70.82	**73.72**

TABLE A.1: 3D results. When SVM is used, we fixed its cost parameter $C = 10$. We performed a nearest neighbors classification with $K = 5$. For H-COV, we used the default choice parameter $L = 3$ with overlap (see [Hus+13]). We selected the ker-COV [Cav+16] parameter after cross-validation.

[Lan+14]. Additionally, in MPII-CAD and Salad 50 for instance, the prediction is facilitated by the detection of *which* objects (out of many others) is grasped (*e.g.* , a knife to predict "cutting"), while, conversely, we want to predict *why* the same object (bottle) is grasped, therefore complicating the applicability of existing prediction pipelines to our problem (Section A.5.1).

A.4 3D motion analysis

Several techniques have been proposed for action recognition from 3D data: bag-of-points [Li+10a], eigen-joints [YT12], Gauss-Markov process [Cha+13], actionlets [Wan+12b], Lie algebra embedding [Vem+14], covariance descriptors [Hus+13], hidden Markov models [LN06], subspace view-invariant metrics [She+05] or occupancy patterns [Wan+12c] to name a few.

Nevertheless, to the best of our knowledge, no attempt has been performed to address the problem of action prediction from 3D data. Thus, in this Section, we will analyse the markers trajectories in our dataset with kinematic features, Dynamic Time Warping and covariance-based representations.

Kinematic Features – Following [Car+11], we computed *wrist velocity*, the module of the velocity of the wrist marker, *wrist height*, the z-component of the wrist marker, *wrist horizontal trajectory* defined as the x-component of the wrist marker and *grip aperture*, *i.e.* the distance thumb-index tips markers. Such features were referred to the motion capture reference system, F_{global} [Car+11]. A better characterization of the dynamics can be provided using a local reference system centered on the hand, F_{local} [Ans+15]. In this way, we computed relative x, y, z coordinates of thumb, index, thumb-index plane and the radius-phalanx. These variables provide the information about either the adduction/abduction movement of the thumb and index fingers or the rotation of the hand dorsum. Thus, they ensure robustness towards finger flexion/extension or wrist rotation that can vary significantly from one trial to another [Ans+15]. The 4 features from F_{global} and the 12 from F_{local} gives a total amount of 16 kinematic features. Acquisition time $[t_0, t_f]$ (see Section A.3) is scaled into $[0, 1]$ and data are sub-sampled with step 0.01. Consequently,

for each of our kinematic features, we have 100 equispaced values describing the evolution of such features during the reach-to-grasp movement: globally, F_{local}, F_{global} and F_K shapes as a 1200, 400 and 1600-dimensional descriptor, respectively, which fed a linear support vector machine (SVM).

Dynamic Time Warping (DTW) – We used DTW to construct a similarity measure Δ between multivariate time-series, exploiting the notion of alignment through warping paths (see [Mül07]). Thus, after computing Δ for all pairs of motion sequences from our dataset, we got the 1098×1098 distance matrix which was both directly used as metric for K-nearest neighbours (K-nn) classification and converted into a kernel by means of the graph Laplacian operator $\mathcal{L}$ to feed SVM classification [Gud+08].

Covariance-based paradigms – We inspected the sampling covariance estimator - briefly, covariance - in predicting human intentions from motion since in the field of action recognition from motion capture (MoCap) systems, many works were actually based on such kind of representation. For instance, [Hus+13] proposed a hierarchical model composed by a L-layered temporal pyramid of covariance descriptors (H-COV). Also, in the recent work [Cav+16], the new state-of-the-art in action recognition from MoCap data as obtained by a rigorous kernelization of the covariance operator (ker-COV) in order to model, general, non-linear, temporal correlations of marker coordinates.

A.4.1 Discussion

In Table A.1, we report the results obtained with all the 3D encodings considered. As expected, when we combine F_{local} and F_{global} in F_K the performance generally improves. Globally, K-nn using DTW is worse than F_K with the exception of Pouring vs. Placing. With respect to the K-nn approach, the graph Laplacian $\mathcal{L}$ allows to boost the DTW classification performance in almost all the binary/all-class comparisons. A further boost in accuracy is provided by H-COV [Hus+13] and ker-COV [Cav+16] with an all-class improvement of 7.72% and 10.62% respectively. Also, in the binary comparisons, with the only exception of Passing vs. Placing and Pouring vs. Passing where the best approaches are DTW and F_K, the scored performance makes ker-COV the best 3D encoding in the all-class comparison and in the remaining ones. Eventually, this has to be read as a strong evidence that a proper modeling of the temporal non-linear correlation of the VICON markers is very effective in predicting IfM. Globally, for all the results in Table A.1, the random chance level (50% in the binary and 25% in the all-class comparisons) is always improved by margin. This demonstrates that the dynamics of the grasping actually encodes some motion patterns which go beyond the bare fulfillment of the action itself and can concretely anticipate the underlying intention. However, since the 3D analysis leverages on a precise and localized temporal evolution of the hand kinematics, in the next Section, we want to investigate the video counterpart of our dataset as to check if, similarly, the random chance in classification can be still overcome.

A.5 Analysis of 2D video sequences

Far from providing a comprehensive review of the whole action recognition/prediction literature on video data, in this Section, we will benchmark the best hand-crafted descriptors (dense trajectories [Wan+13c]) as well as 3D Convolutional Neural Network (CNNs) for video representation and frame-based deep encodings for our graspings. Moreover, we will show that many currently available frameworks in early activity recognition and action prediction are not suitable for our test-bed problem.

Dense trajectories (DT) – Being part of the class of approaches named in [AR11] as local, DT [Wan+13c] track in time a set of spatio-temporal interest points (IPs) from an input video, using a dense optical flow field. For each IP, its trajectory is surrounded by a warped volume from which we computed classical histogram features: Histograms of Oriented Gradients (HOG) [DT05], Histograms of Optical Flow (HOF) [Cha+09b], Motion Boundary Histograms [Dal+06] in both x and y directions (MBHx and MBHy), trajectory shape descriptor (TSD) [Wan+13c] and Histograms of Oriented Trackelets (HOT) [Mou+15]. We used the publicly available DT code[2], adopting the default parameters choice except to the trajectory length which was set to 5 to better deal with our extremely short footages.

In order to combine the dense histogram features into a unique video descriptor, we either applied ℓ^1 normalized bag-of-features histograms (*BoF*) [Boi+08], square-root normalized Fisher Vector (*FV*) [PD07], or Vectors of Locally Aggregated Descriptors (*VLAD*) [J+10]. For *BoF* and *VLAD*, we used a dictionary of 1000 visual words; for *FV* we employed a Gaussian mixture model with 256 components (as in [WS13]).

CNN features – We applied the three-dimensional convolutional network architecture C3D proposed in [Tra+15]. Thus, we divided each video sequence in three clips of 16 frames, where each of them is codified with `fc6` features. The video descriptor simply concatenates the three representations of the clips into a 3×4096 vector, finally used to train a linear SVM. As input clips, we have considered stacks of raw frames (I), also representing the optical flow (OF) magnitude computed between pairs of consecutive frames.

Similarly to [Ng+15; Zha+15a] we exploited CNNs for a frame-wise representation, employing, as a first setup, the AlexNet architecture once fine-tuned on our video frames. Precisely, we extracted `fc7` features from all single frames I and, consequently, we encoded each video with *BoF* as in [Xu+15b]. In a second experiment, in order to better capture the kinematics of the graspings, we fed AlexNet with OF images after another preliminary fine-tuning to match the new type of data. Inspired by [Ché+15], in this case, we computed OF images with three channels constituted by the horizontal, vertical component and the magnitude of the optical flow field, after a preliminary normalization in the range $[0, 255]$. To obtain the descriptor for each video, we either applied *BoF* and *VLAD* encoding upon the AlexNet-OF deep representation.

[2] http://lear.inrialpes.fr/software/

DT results	HOG (%)		HOF (%)		TSD (%)	
Pouring	*BoF*	85.28	*BoF*	**85.75**	*BoF*	83.23
vs.	*FV*	87.12	*FV*	**87.59**	*FV*	78.84
Placing	*VLAD*	86.71	*VLAD*	86.18	*VLAD*	81.60
Pouring	*BoF*	70.63	*BoF*	**77.21**	*BoF*	72.66
vs.	*FV*	75.48	*FV*	**81.03**	*FV*	73.39
Drinking	*VLAD*	77.33	*VLAD*	**81.48**	*VLAD*	77.92
Pouring	*BoF*	71.77	*BoF*	67.41	*BoF*	72.17
vs.	*FV*	75.90	*FV*	**79.75**	*FV*	67.16
Passing	*VLAD*	**77.48**	*VLAD*	74.44	*VLAD*	73.58
Passing	*BoF*	66.67	*BoF*	65.49	*BoF*	65.34
vs.	*FV*	66.88	*FV*	**73.22**	*FV*	61.44
Drinking	*VLAD*	70.06	*VLAD*	**71.53**	*VLAD*	67.68
Passing	*BoF*	**66.99**	*BoF*	66.34	*BoF*	58.28
vs.	*FV*	65.55	*FV*	**76.84**	*FV*	56.00
Placing	*VLAD*	65.83	*VLAD*	**75.15**	*VLAD*	61.62
Drinking	*BoF*	70.66	*BoF*	**75.60**	*BoF*	68.77
vs.	*FV*	73.04	*FV*	**78.41**	*FV*	73.99
Placing	*VLAD*	72.55	*VLAD*	**79.23**	*VLAD*	72.24
	BoF	**48.16**	*BoF*	48.05	*BoF*	45.01
All-class	*FV*	50.02	*FV*	**56.97**	*FV*	46.00
	VLAD	51.88	*VLAD*	**58.23**	*VLAD*	51.63

DT results	MBHx (%)		MBHy (%)		HOT (%)	
Pouring	*BoF*	85.56	*BoF*	84.64	*BoF*	83.64
vs.	*FV*	85.96	*FV*	83.06	*FV*	78.64
Placing	*VLAD*	**87.65**	*VLAD*	85.70	*VLAD*	76.93
Pouring	*BoF*	70.78	*BoF*	74.15	*BoF*	64.81
vs.	*FV*	76.73	*FV*	75.46	*FV*	62.40
Drinking	*VLAD*	74.61	*VLAD*	76.22	*VLAD*	60.23
Pouring	*BoF*	**72.55**	*BoF*	68.33	*BoF*	72.22
vs.	*FV*	75.15	*FV*	70.17	*FV*	65.58
Passing	*VLAD*	76.01	*VLAD*	68.48	*VLAD*	63.14
Passing	*BoF*	**71.21**	*BoF*	64.65	*BoF*	69.10
vs.	*FV*	68.45	*FV*	68.02	*FV*	64.69
Drinking	*VLAD*	69.78	*VLAD*	66.25	*VLAD*	61.02
Passing	*BoF*	65.25	*BoF*	59.57	*BoF*	66.64
vs.	*FV*	66.86	*FV*	60.02	*FV*	63.75
Placing	*VLAD*	67.85	*VLAD*	63.80	*VLAD*	62.63
Drinking	*BoF*	73.19	*BoF*	71.60	*BoF*	70.44
vs.	*FV*	74.04	*FV*	73.04	*FV*	65.05
Placing	*VLAD*	75.35	*VLAD*	73.27	*VLAD*	63.84
	BoF	47.71	*BoF*	45.80	*BoF*	46.33
All-class	*FV*	50.35	*FV*	47.23	*FV*	41.12
	VLAD	53.30	*VLAD*	48.63	*VLAD*	38.62

TABLE A.2: DT features for SVM classification (C = 10). For *BoF*, we computed an exponential χ^2 kernel, while, for *FV* and *VLAD*, a linear kernel was adopted.

CNN results (%)	C3D		AlexNet		
	I	OF	I	OF	
			BoF	*BoF*	*VLAD*
Pouring vs. Placing	83.15	87.34	74.01	**94.58**	94.18
Pouring vs. Drinking	68.20	68.93	62.44	74.93	**77.95**
Pouring vs. Passing	69.42	65.02	60.28	**75.89**	74.20
Passing vs. Drinking	61.57	61.05	55.73	61.41	**66.05**
Passing vs. Placing	66.84	78.10	62.09	**96.23**	94.68
Drinking vs. Placing	68.30	76.67	58.69	95.87	**96.18**
All-class	45.51	52.14	37.03	64.55	**65.64**

TABLE A.3: Accuracies of pre-trained C3D and fine-tuned AlexNet architectures used as feature extractors for a subsequent SVM classification (C = 10). When using *BoF* and *VLAD*, the size of the dictionary was fixed to 1000 and 50 respectively.

A.5.1 Discussion and evaluation of existing prediction pipelines

In Table A.2 and A.3, we report the 2D classification results related to DT and deep features, respectively.

Among all the inspected histogram representations in Table A.2, in general, HOT provide the lowest classification results across the different comparisons, no matter which *BoF*, *FV* and *VLAD* high level encoding was used on top. Instead, HOG, HOF, TSD and MBH histograms show a comparable performance, even if it is worth nothing that HOF frequently provide the best accuracy in most comparisons.

Moving to the CNN results, while comparing the two different inputs we used (I and OF), C3D architecture registers an improved performance when using the latter. Nevertheless, in the all-class case, C3D+OF (52.14%) registers a slightly inferior performance to the DT-HOF-*VLAD* (58.23%). Even though CNNs are the best known approach for image classification to date, the suboptimal results of AlexNet-I-*BoF* assess that a static analysis of the frames of our dataset is not so effective. Actually, what really matters is the motion information which can better be captured by AlexNet-OF-*BoF* and AlexNet-OF-*VLAD*. Precisely, the latter representation scores the highest 2D accuracy on the all-class case. Also, the use of the same features obtains very high results (>94%) on all the binary classification which involves the Placing intention, which, surprisingly, turns out to be easily discriminated by means of CNNs.

In some binary comparisons (*e.g.* , Pouring vs. Drinking), the 3D baseline shows a superior performance with respect to the 2D case. Then, in order to improve the latter results and to bridge the gap in performance with the 3D case, instead of action recognition techniques applied to a prediction task, we could directly apply existing action prediction pipelines for IfM. Hopefully, this will lead to gain in performance.

Despite the broad literature in action prediction and early activity recognition, most of approaches are not directly applicable to the IfM problem. Indeed, [Lan+14] relies on a fine decomposition of the action into coarse, mid-level and fine actions classes: of course, this is not applicable to our simple grasping

Fusion results (%)	BSD	*Early Fusion*		*Late Fusion*	
		PCA	CMIM	MSE	ACC
Pouring vs. Placing	94.18	85.80	**95.92**	88.84	95.70
Pouring vs. Drinking	91.58	83.85	93.30	91.41	**94.62**
Pouring vs. Passing	84.47	79.88	**90.47**	85.85	90.04
Passing vs. Drinking	87.75	84.89	87.21	82.57	**90.30**
Passing vs. Placing	**96.23**	70.23	93.49	82.33	91.12
Drinking vs. Placing	96.18	82.63	93.68	90.97	**96.87**
All-class	73.72	68.39	80.08	77.52	**80.50**

TABLE A.4: Early fusion of feature descriptors and late fusion of kernels.

movements. [Von+16] relies on a convolutional network which is trained by jointly considering the present and the future of a given scene, while, in our case, only the (present) graspings are exploitable as data. Despite [KS13] deals with grasping motor acts as we do, it only predicts *which* object is grasped, not *why*, as we aim at. Finally, [HD12] and [Li+12] need massive annotations of the emotion disclosure and actionlets respectively, while, in this sense, our problem is fully unsupervised.

Among the few works directly applicable to our problem, we evaluate the temporal tessellation and dynamic bag-of-word histograms proposed by [Ryo11]. Using this algorithm, the all-class classification accuracy results 45.12%, which suffers a gap of −13.11% and −20.52% with respect to DT-HOF-*VLAD* and AlexNet-OF-*VLAD*, respectively. Thus, globally, despite all the aforementioned prediction pipelines are really effective in their experimental conditions, the same methods seem little generalizable to different settings (such as ours).

In the end, despite the broad 2D analysis presented, the 3D data processing baseline scored a superior performance (in terms of classification accuracy), and to bridge such gap, we will prove in the next Section that a multi-modal data fusion is beneficial in this sense.

A.6 Fusing 3D and 2D information

A unique aspect of our proposed dataset refers to its multi-modal nature, namely providing both 3D markers trajectories and 2D video acquisitions of every reach-to-grasp onset. Thus, it is interesting to take advantage of such dual source of information to overcome the performance of simpler methods which only leverage on one type of data only. To this aim, in this Section, we present some baseline experiments as to combine all the inspected 3D and 2D feature representations in a unique descriptor, performing an *early fusion* of our data. Further, we investigate some basic *late fusion* modalities, where, at a higher level, we performed a combination of kernels, each of them representing each feature encoding separately. For a comprehensive analysis on fusion techniques, please refer to [Atr+10].

Early fusion of feature vectors – Throughout our 3D and 2D baseline, several features have been envisaged: F_K, ker-COV [Cav+16], the six DT histogram descriptors and the deep representations extracted by either using C3D

3D snippet analysis	20%	40%	60%	80%	100 %
Pouring vs. Placing	51.48	71.40	76.89	87.55	**91.87**
Pouring vs. Drinking	47.85	61.01	64.45	72.30	**91.58**
Pouring vs. Passing	47.89	52.75	56.98	70.77	**81.69**
Passing vs. Drinking	50.75	53.98	59.62	71.97	**87.64**
Passing vs. Placing	54.20	61.67	62.75	70.68	**75.46**
Drinking vs. Placing	54.48	60.00	64.93	67.84	**91.24**
All-class	27.90	33.31	38.60	49.03	**73.72**

TABLE A.5: Results for the snippet analysis using ker-COV features.

2D snippet analysis	20%	40%	60%	80%	100 %
Pouring vs. Placing	77.38	87.02	91.43	92.06	**94.18**
Pouring vs. Drinking	61.31	67.49	71.52	73.31	**77.95**
Pouring vs. Passing	57.82	65.47	66.00	69.98	**74.20**
Passing vs. Drinking	61.25	62.85	65.25	65.62	**66.05**
Passing vs. Placing	90.93	96.19	95.82	**96.28**	94.68
Drinking vs. Placing	89.42	95.00	95.42	95.18	**96.18**
All-class	49.70	57.79	57.91	62.27	**65.64**

TABLE A.6: Results for the snippet analysis using AlexNet-OF-*VLAD* features.

or AlexNet. In order to fuse all of them into a unique descriptor, we applied two techinques.

1. We concatenated all the aforementioned feature representations into a unique single vector and reduced its dimensionality from 579.786 to 160 components by means of PCA (38% variance explained).

2. We applied the CMIM criterion [Bro+12] to capture the variability in the class label conditioned on the data, while also minimizing the redundancy with respect to previously selected component (see [Bro+12]). In our case, we used CMIM to select the 150 most discriminative feature components among all the different single representations from Sections A.4 and A.5.

Late fusion of kernels – As the preliminary stage of our late fusion pipeline, we computed a kernel from each different data encoding separately: a Gaussian RBF kernel for each kinematic feature, the graph Laplacian for the DTW similarity matrix, a Gaussian χ^2 kernel for AlexNet-I-*BoF* and AlexNet-OF-*BoF*. A linear kernel was used for ker-COV, C3D-I, C3D-OF and for the DT, AlexNet-OF features encoded with *VLAD*. In order to train a SVM, the final kernel used is a linear combination of all the aforementioned ones, weighted according to the MSE and ACC criteria proposed in [Atr+10]. That is, each kernel is weighted according to the mean squared error (for MSE) and to the classification performance (for ACC) registered when using a SVM fed with that single kernel only.

A.6.1 Discussion

The classification accuracies are reported in Table A.4, where we also include the best single descriptor (BSD) among all the ones presented in Sections A.4 and A.5. The early fusion is able to improve the performance of BSD on Pouring vs. Placing and Pouring vs. Passing. Similarly, with the exception of Passing vs. Placing, the late fusion improves all the remaining pairwise comparisons. Moving to the all-class case, the late fusion (ACC) and early fusion (CMIM) behaves in a similar manner, both overcoming the 80% of classification accuracy. Thus, in addition to certify that the different feature representations capture complementary aspects of the problem, the fusion methods produce classification scores in the all-class case which improves the random chance level by about 55%.

As a consequence, we want to better understand what precisely went on at the classification stage, as to more carefully inspect our case study. This is done in the next Section, where, additionally, we will devise an automatic pipeline to actually solve the classification problem in our test-bed dataset.

A.7 Personalizing Human Intention Prediction

Throughout the classification analysis performed over the 3D marker joints trajectories (Section A.4), the 2D video sequences (Section A.5) and the multimodal fusion (Section A.6), we assessed the feasibility of predicting IfM. Indeed, despite the apparent similarity of the same grasping performed with different intentions (Figure A.2), the registered classification accuracies improved by margin the random guess level. Consequently, we deem interesting to investigate why such results were actually obtained. That is, we tried to mine the subtle differences which discriminate the intentions, also understanding what is the main issue that complicates the prediction.

To this aim, in Section A.7.1, we will first perform a temporal inspection of our grasping actions, searching for any instant which is more informative than others. In Section A.7.2, we will gain an improved knowledge of which feature components were actually discriminant for each intention. As a consequence of a subject-detailed analysis (Section A.7.3), we will be able to envisage a novel action prediction pipeline properly tailored for our case study, so that we solved the classification in our proof of concept for IfM (Section A.7.4).

A.7.1 Snippet Analysis

In this Section, we present a temporal analysis of the reach-to-grasp motions to verify if, in the 3D/2D data, it is possible to find any peculiar instant which is richer in kinematic discriminants than others.

We performed a snippet analysis where the 3D marker trajectories and 2D video sequences were trimmed to cover the initial 20%, 40%, 60%, 80% or 100% of the original grasping execution only. Please note that, on our dataset, the latter is extremely short (2 seconds on average): hence, the snippet analysis forces the classification to rely on a very limited information. For instance,

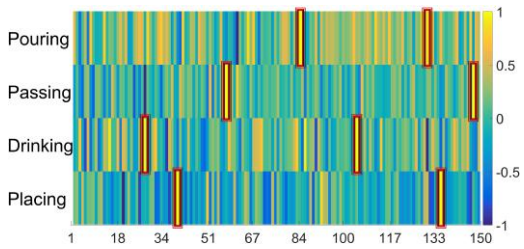


FIGURE A.4: Global, one-intention-versus-the-rest SVM weights for CMIM feature selection. A yellow (resp. blue) color means that the corresponding component is highly (resp. poorly) discriminative for the specific intention.

at 20%, for the shortest trial in our dataset, we have to use 16 markers acquisitions and 3 video frames only. As to monitor the impact of limiting the temporal domain on the 3D and 2D data separately, we considered the descriptors which obtained the best performance in the two baselines, respectively: ker-COV [Cav+16] (Section A.4) and AlexNet-OF-*VLAD* (Section A.5). In this case, ker-COV representation only models temporal correlations among the initial 20%, ..., 100% acquisitions of the markers and, similarly, the dictionary for *VLAD* encoding AlexNet-OF features is specific of the considered portion of the videos only.

Table A.6 report the results of the snippet analysis for AlexNet-OF-*VLAD*. Therein, the best scores are obtained by considering high percentages (80% and 100%), but, anyway, we are able to capture discriminative information already from the beginning of the grasping: *e.g.* 90.93% in Passing vs. Placing at 20 %. Differently, in Table A.5, the snippet analysis with ker-COV does not remarkably exceed random chance level at 20% and 40%, with a great jump in performance at 80% and 100%.

In spite of this, we can anyway find a common trend between the results of Tables A.5 and A.6. Namely, we registered a general growth in accuracy when the data percentages increase. Consequently, we can not find any portion of the reach-to-grasp execution that is completely useless for the prediction of intentions. This signifies that, in our proof of concept, the discrimination of IfM is done by accumulating kinematic differences across the whole execution of the graspings.

Since we certified that a temporal segmentation is not beneficial, in the next Section, we will be interested in mining which features turned out to be maximally discriminant for each single intention.

A.7.2 Mining the intention discriminants

In this Section, we are interested in highlighting the intention-specific differences that are actually exploited during the classification stage. Hence, we focused on the best feature representation obtained through the CMIM[3] early fusion pipeline (Section A.6). As previously explained, CMIM provided

[3]Properly, the late fusion results are slightly better than the early ones. However, since the approach is not providing an explicit feature encoding (but only a kernel), we actually consider the CMIM technique.

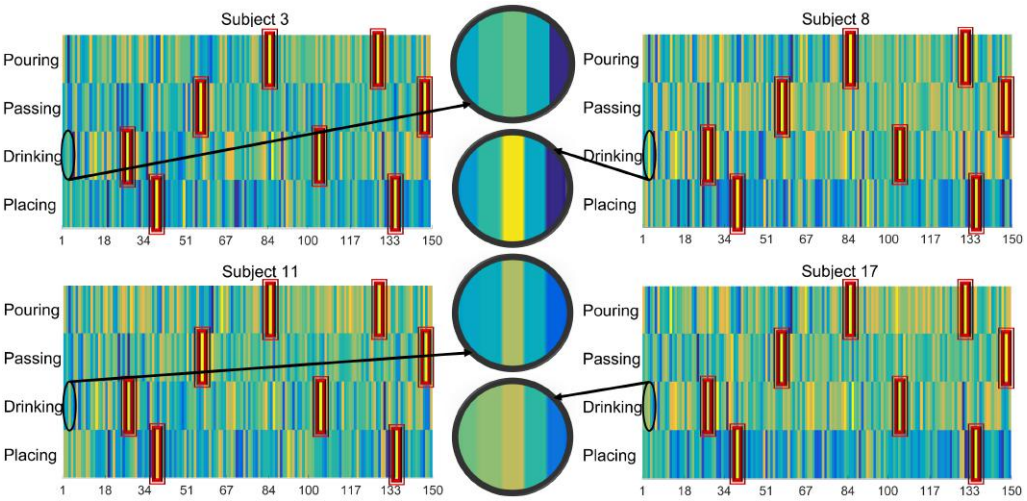


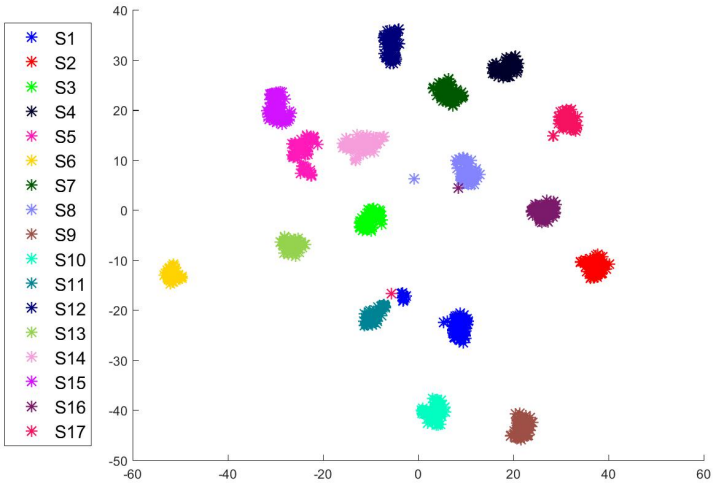
FIGURE A.5: Subject-specific, one-subject-out SVM weights for CMIM feature selection (subjects 3, 8, 11 and 17). A yellow (resp. blue) color means that the corresponding component is highly (resp. poorly) discriminative for the specific intention.

the best 150 feature components which maximally allow to disambiguate between our four classes (intentions), guaranteeing a low-redundant representation [Bro+12]. Fixing such multi-modal data encoding, we tackle a class specific investigation finalized to recognize which of those selected components are more proficient than others in characterizing any particular intention.

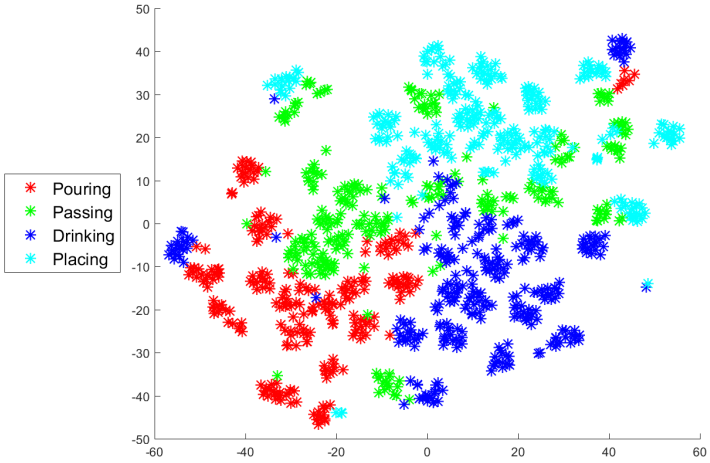
To this aim, we inspected the weights of a linear SVM trained in a one-intention-versus-the-rest fashion, involving the whole dataset, where the data examples referring to one intention are positive and those referred to the other 3 intentions are the negative examples. After filtering out (by setting to zero) the weights with absolute value lower than a fixed threshold (10^{-3}), for each SVM model, we linearly scaled the remaining weights to fall within the range $[-1, 1]$. The results of such a pipeline are reported in Figure A.4, where the color map depicts in yellow/blue the entries of the SVM weights which are greater in magnitude and positive/negative in sign, clearly corresponding to those feature components which are interpreted by the SVM to be the most/least discriminant for that particular intention. Furthermore, if, for the same component, its weight is high for *one* intention and low in *all* the other cases, it means that all the four SVM models agree in recognizing that such component is specific for one intention *only*. Thus, in Figure A.4, the red bounding boxes specify which component (column) is maximally discriminant for which intention (row).

For the sake of fairness, it is worth nothing that, for the SVM training, the one-intention-vs-the-rest strategy is different if compared to the one-subject-out adopted before. Despite Figure A.4 leads to a global statistics, it is better to check what changes if we consider the exact SVM models which were used to predict the intention for each of the 17 subjects in our baseline results.

Thus, in the same manner, we filtered and scaled into $[-1, 1]$ the weights of the 17 multi-class, linear SVM models where each subject is left out for testing. So, let's consider the weights associated to 4 (for brevity) randomly chosen subjects in Figure A.5. Therein, it is evident that each subject's model shows a similar trend if compared to Figure A.4 - for instance, the two components



(a) t-SNE using F_K – each color represents one subject.



(b) t-SNE using CMIM selected features – each color represents one intention.

FIGURE A.6: Bi-dimensional embedding using t-Distributed Stochastic Neighbor Embedding for the proposed test-bed dataset for IfM.

highlighted in red for Pouring are still brightly colored. However, after a more careful insight, Figure A.5 shows that each subject has also his/her own peculiarities. In fact, zooming in (centered circles in the figure) the same component (bounded by the black ellipses) related to the Drinking-vs-the-rest SVM models for the 4 subjects, one can note a bright yellow coloration for one subject only. Specifically, the highlighted component is highly Drinking-specific for Subject 8, being at the same time not so relevant for Subject 3, 11 and 17. This certifies that, in addition to the subject-generic and intention-specific discriminants (Figure A.4), there also exist some intention-specific patterns which are relevant for specific subjects only (Figure A.5). As a consequence, in the next Section, we will take care of better understanding the role played by the subjects in our intention prediction problem.

A.7.3 The role of the subjects in IfM

In all the baseline experiments presented, the adopted one-subject-out testing modality forces the classifier to predict the intention of a human actor which was never used in the training phase. Therefore, the classification task must rely on some intention-anticipating discriminant clues, which, at the same time, have to be shared across the subjects. Except a few cases [YVG14], the role of the subject has been frequently disregarded and sometimes underestimated for the task of action and activity recognition or prediction. In this Section, we want to investigate such a role in our one-subject-out testing modality, namely checking if the unknown identity of the subject who is acting complicates the prediction of his/her intention. As a relevant collateral effect, such improved understanding helps us in devising an original action prediction method (described in Section A.7.4) which is tailored for our dataset, greatly boosting the prediction performance.

To fully understand the impact of the subject, we first analyzed the data to find how much subject-dependent information they convey. Actually, such aspect is not useful for IfM, being a sort of noise which can potentially affect the performance. Hence, we evaluated such trait on the kinematic features F_K since they are the representation which more directly codifies the (3D) data (see Section A.4), providing a low level encoding if compared, for instance, with AlexNet-OF-*VLAD*. Despite F_K are always able to exceed the random chance in all the comparisons, their performance is suboptimal if compared with other feature representations. Actually, such aspect can be explained by considering that F_K tend to capture more subject-related rather than intention-related kinematic cues. Indeed, such interpretation is confirmed by Figure A.6(a), where we applied the t-Distributed Stochastic Neighbor Embedding (t-SNE) technique [MH08] to infer an approximating embedding from a high-dimensional feature encoding. In this way, we register the presence of several compact and distinct clusters. It is worth noting that t-SNE is an unsupervised technique which does not exploit neither the subject identity nor the intention label at all. However, as a post-processing stage done on the two-dimensional t-SNE embedding of F_K, we colored the obtained 2D points as to represent which subject performed what trial. So, we get the net result that each cluster precisely corresponds to one specific subject, no matter which is the underlying intention. Actually, Figure A.6(a) states that, in F_K, the subject-specific information overcomes the intention-anticipating patterns, being the latter nevertheless present (55.13% scored by F_K in the all-class comparison).

Figure A.6(a) can be read also in a more formal way, observing that, once considering the data of each intention separately, the intra-subject variability is much lower than the inter-subject variability. Moreover, two generic elements of the same class (intention), but different subjects, are more far away than two generic instances of the same subject which grasps the bottle with two different intentions. Hence, what actually defines a good feature representation for our proof of concept dataset is 1) enhancing the subtle kinematic differences between the intentions while, at the same time, 2) bridging the differences across subjects that we found in Figure A.5. Thus, an efficient feature representation should be able to better cluster the data in a way that the four intentions occupy 4 separate regions in the feature space. Again, such hypothesis can be

confirmed by the usage of the t-SNE embedding applied to the CMIM representation, that is the best descriptor in terms of classification accuracy. The results are reported in Figure A.6(b), where an ordered structure can be seen: in the approximated low dimensional feature space, we found 4 quite detached groups, each of them semantically corresponding to one intention.

A.7.4 Two-layer SVM architecture for IfM

In Sections A.7.2 and A.7.3, we realized that the role of the subject is actually important. By taking advantage of that, we can tackle the classification problem in our attempt for IfM by considering a *divide-et-impera* setting where we split the original (more complicated) problem into several sub-problems, each of them turning out to be more easily solvable. Actually, in our case, each of the sub-problem precisely corresponds to a single subject at a time. Inspired by our experimental outcomes, we claim that the intention prediction task is easier given the identity of the actor who is grasping the bottle. Indeed, in this way, more kinematic cues can be exploited: in addition to the intention-specific and subject-generic kinematic discriminants (Figure A.4), in each sub-problem related to any single subject, the classifier can also benefit from the intention- and subject-specific patterns we retrieved in Figure A.5.

In order to cast this idea into a computer vision pipeline, actually customized to the prediction of the intention from human motion only, we propose the following novel approach based on a two-layer SVM architecture (as shown in Figure A.7).

Layer 1 – We performed a preliminary subject identification stage using a linear SVM, trying to recognize who is actually grasping the bottle by just inspecting how the grasping itself is performed. Despite such task seems very hard a priori, Figure A.6(a) actually suggests that the kinematic features F_{K} could fruitfully discriminate the subject.

Layer 2 – We predicted IfM in the simplified setting where, as classifier, we applied a specific SVM model trained over the samples belonging to the recognized subject only. Taking into account the experimental findings of Section A.7.2, once the subject identity is known, we can take advantage of both the subject-generic (Figure A.4) and the subject-specific (Figure A.5) kinematic discriminants which are useful to predict intentions. Thus, in the second layer, it seems natural to adopt the CMIM feature representation that we used in that analysis. In addition, the reduced dimensionality of CMIM features is particularly suited to prevent the curse of dimensionality issue [Bis06] related to the limitedness of the number of the available training samples in the case of each single sub-problem.

Despite a few previous works proposed a network of SVMs for image classification [Tan13] and general regression tasks [Wie+13], our approach (for intention prediction) turns out to be original. Indeed, in [Tan13], a SVM replaces the final softmax layer of a CNN, while, differently, our network is fully composed by SVMs. Also [Wie+13] proposed a two-layer network of SVMs where, in the first layer, the scores produced by a stack of SVMs are passed as input to a unique SVM which performs the final classification. Our model differs from

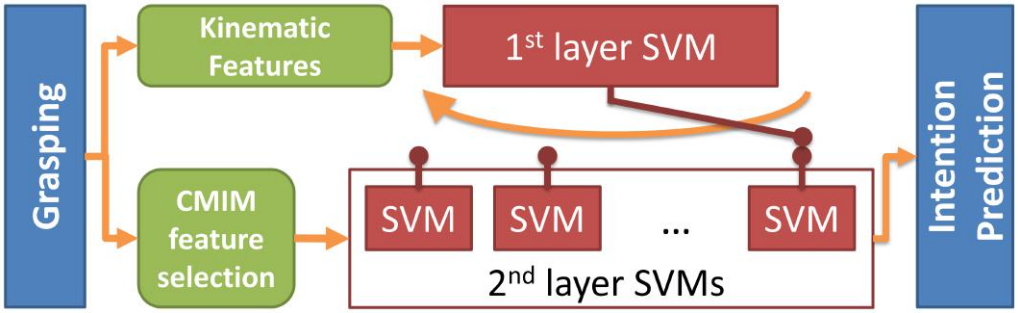


FIGURE A.7: Outline of the proposed two-layer SVM architecture.

[Wie+13] in two aspects. First, as a reversed architecture, we set one SVM in the first layer and multiple ones in the other. Second, each layer of our network is fed with an independent data representation.

In our dataset, such composite two layers pipeline is very appealing. Indeed, Figure A.6(a) states that F_K seems very efficient in codifying the subject identity, albeit being not the best descriptor in terms of accuracy to predict intentions. Conversely, the CMIM feature provided a top accuracy in IfM, while, at the same time, such representation is obtained after a too sophisticated encoding, actually loosing all subject information. By exploiting F_K to recognize the subject and CMIM to perform the intention prediction, in a few words, our two layers SVM pipeline collects the pros of the two representations.

Implementation details. In order to train our two-layer SVM architecture, we adopted the following scheme. First, we performed a random 2/3-1/3 partition of the whole dataset in training-test set, respectively, once being sure to perform such splitting in a balanced way across both subjects and intentions. As to train the 1^{st} layer SVM, we used as data representation F_K and as label the ground truth subject identity of who is the actor which grasped the bottle in that particular trial. Afterwards, we split the same training set used before in 17 sub-parts, each of them collecting the grasp-to-Pour, grasp-to-Pass, grasp-to-Drink and grasp-to-Place examples of one single subject at a time. Then, focusing on each sub-part separately, we trained the 2^{nd} layer SVMs as intention predictors: being specific of each subject, these multiple SVM models used the CMIM feature as descriptor and the ground truth intention label. Globally, once the second layer is trained, we obtained 17 subject-specific SVM intention predictors.

At the testing stage, the same trial is codified with both F_K and CMIM: F_K is used to obtain the 1^{st} layer SVM prediction, estimating the subject identity. The latter routes to the specific SVM model within the 2^{nd} layer SVMs which refers to the recognized agent. That specific classifier is fed with CMIM and produced the final prediction of the intention which underlies each testing sample.

Performance evaluation. In order to have a quantitative evaluation of our proposed pipeline, we benchmarked the performance of each layer of the network separately. To ensure robustness within the training/test splitting, we repeated such random partition 20 times, thus reporting the mean accuracy values in Table A.7.

1^{st} layer SVM	
Subject classification	98.68%
2^{nd} layer SVMs	
Pouring vs. Placing	**99.89%**
Pouring vs. Drinking	**98.63%**
Pouring vs. Passing	**98.49%**
Passing vs. Drinking	**97.95%**
Passing vs. Placing	**98.83%**
Drinking vs. Placing	**99.32%**
All-class	**97.37%**

TABLE A.7: Two-layer SVM architecture performance. We separately reported the 1^{st} layer SVM in recognizing the subject identity as well as the 2^{nd} layer SVMs accuracies in predicting the actual intention.

From such results, it is evident that, in the first layer, the performance of F_K in 1^{st} layer SVM is capable of providing a high classification of the subject's identity. Grounding on this, the routing criteria to the subject-related SVM in the 2^{nd} layer SVMs is therefore very precise: with a high degree of confidence, the intention prediction stage is done by applying the SVM model which is correctly trained on the instances of the same actor to which the testing example belongs.

Furthermore, when inspecting 2^{nd} layer SVMs results, we greatly improved the best performance ever scored by any method throughout Sections A.4, A.5 and A.6: on average, we registered +3.97% in Pouring vs. Placing, +4.01% in Pouring vs. Drinking, +8.02% on Pouring vs. Passing, +7.65% in Passing vs. Drinking, +2.60% in Passing vs. Placing, +2.45% in Drinking vs. Placing and **+16.87%** in the all-class case.

A.7.5 Validation on classical action recognition datasets.

Inspired by our findings we discovered in this Section, we posit that such a two-stage personalization pipeline may be effective also when dealing with classical action recognition benchmarks.

Our investigation involves three publicly available MoCap datasets for activity recognition: MSR-Action3D, MSRC-Kinect12 and HDM-05. In all our experiments, we only used the 3D skeleton coordinates while the other data available (*e.g.*, depth maps or RGB videos) were not considered. For the sake of clarity, we briefly introduce each of them.

— **MSR-Action3D** [Li+10b] dataset has 20 action classes of mostly sport-related actions (*e.g.*, *jogging* or *tennis-serve*), performed by 10 subjects. $J = 20$ joints are extracted from the Kinect sensor data to model the human pose of the human agents. Each subject performs each action 2 or 3 times. In total, we used 544 sequences [Hus+13].

— **MSRC-Kinect12** [Fot+12] is a relatively large dataset of 3D skeleton data, recorded by means of a Kinect sensor. The dataset has 5881 sequences, containing 12 action classes performed by 30 different subjects. Each subject

accomplishes each class of action 16 times, on average. The available motion files contain the trajectories estimated for $J = 20$ 3D skeleton joints.

— In **HDM-05** [M̈+07], the number of skeleton joints is $J = 31$, each action is repeated 5 times on average by each of the 5 subjects involved during the acquisition through a VICON system. We followed the 14-classes experimental protocol of [Hus+13; Roz+16].

For all the aforementioned datasets, each trial can be formalized as a collection $\mathbf{S}$ of τ different acquisitions $\mathbf{p}(1), \dots, \mathbf{p}(\tau)$. For any $t = 1, \dots, \tau$, we denote with $\mathbf{p}(t)$ the column vector which stacks $\mathbf{p}_1(t), \dots, \mathbf{p}_J(t) \in \mathbb{R}^3$, the three-dimensional x, y, z coordinates of the J skeletal joints. Using this notation, we now briefly introduce the two different representations for MoCap data.

First, we investigated the usage of dynamic time warping (DTW), a classical tool to quantify the similarity across two different time series by means of alignment [Mül07; Gud+08]. In order to apply DTW, we evaluated the differences between any two joints collection $\mathbf{S} = [\mathbf{p}(1), \dots, \mathbf{p}(\tau)]$ and $\mathbf{S}' = [\mathbf{p}'(1), \dots, \mathbf{p}'(\tau')]$ through the following distance

$$d(\mathbf{p}(s), \mathbf{p}'(t)) = \frac{1}{J} \sum_{j=1}^{J} \|\mathbf{p}_j(s) - \mathbf{p}'_j(t)\|, \tag{A.1}$$

where $\|\cdot\|$ is the Euclidean norm, $s = 1, \dots, \tau$ and $t = 1, \dots, \tau'$. The final similarity measure, provided by DTW to compare $\mathbf{S}$ and $\mathbf{S}'$, is $\delta(\mathbf{S}, \mathbf{S}')$ which is the minimum value of (A.1) computed over all the sequences of timestamps which optimally align $\mathbf{S}$ with $\mathbf{S}'$ (see [Mül07] for more details).

Second, we also estimated the $n \times n$ covariance matrix

$$\mathcal{C} = \frac{1}{\tau - 1} \sum_{t=1}^{\tau} (\mathbf{p}(t) - \bar{\mathbf{p}})(\mathbf{p}(t) - \bar{\mathbf{p}})^\top, \tag{A.2}$$

related to any trial $\mathbf{S}$, where $\bar{\mathbf{p}} = \frac{1}{\tau} \sum_{s=1}^{\tau} \mathbf{p}(s)$ averages all the τ coordinates and we denote $n = 3J$ for convenience. Since $\mathcal{C}$ is positive definite, we thus exploited the theory of the Riemannian manifold Sym_n^+ and projected (A.2) onto the tangent space to obtain $\widetilde{\mathcal{C}}$ [Ars+06]. Then, using the symmetry of $\widetilde{\mathcal{C}}$, we extracted its independent entries, yielding the following $n(n+1)/2$ vector

$$\mathrm{COV} = [\widetilde{\mathcal{C}}_{11}, \dots, \widetilde{\mathcal{C}}_{1n}, \widetilde{\mathcal{C}}_{21}, \dots, \widetilde{\mathcal{C}}_{2n}, \dots, \widetilde{\mathcal{C}}_{nn}]. \tag{A.3}$$

Note that the usage of covariance is inspired by [Roz+16], which set the new state-of-the-art performance for action recognition from MoCap data. Also, our approach is similar to the case $L = 1$ in [Hus+13], where a L-layered temporal hierarchy of covariance descriptors is proposed, but differently from us, the projection stage onto the tangent space is not considered.

For both representations, we used the support vector machine[4] (SVM) for classification: when fed with COV, we normalized the data imposing zero mean and unit variance and we then used a linear kernel. Instead, the negative dynamic time warping kernel function [Gud+08] produced the training and testing Gram matrices given in input to the SVM.

[4] In all experiments, for the SVM cost parameter, we fixed $C = 10$.

For the previous two encodings, we report the action recognition classification results in Tables A.8, A.9, respectively. Also, we provide the mean and standard deviation of the accuracies scored in the two steps separately, over 20 different random partitions of the data.

	MSR-Action3D	MSRC-Kinect12	HDM-05
subject-SVM	90.74 ± 2.41	85.18 ± 0.55	85.67 ± 3.18

TABLE A.8: Two-stage recognition pipeline - subject identification accuracies.

	MSR-Action3D	MSRC-Kinect12	HDM-05
action-SVMs	90.46 ± 1.17	$\mathbf{97.14 \pm 0.39}$	97.03 ± 1.36
SoA	**96.9** [Roz+16]	95.0 [Cav+16]	**98.1** [Cav+16]

TABLE A.9: Two-stage recognition pipeline - action classification accuracies.

Discussion. Since COV is designed for action recognition, it is suboptimal for subjects' identification. In fact, despite the classification performance we registered is still reliable (Table A.8), when a subject is misclassified, the action classifier corresponding to another subject is used and performance can deteriorate.

Nevertheless, we only registered a 2% the drop with respect to *Personalization* strategy, which can be considered as our two-stage pipeline with perfect subject recognition in the first stage. Such performance is remarkable since, after all, *Personalization* requires the subjects' identity to be known, whereas we are effectively able to automatically learn it [5].

Although a comparison of our simple approach with more sophisticated approaches [Roz+16; Hus+13; Cav+16] is challenging, we score a favorable performance with respect to the state-of-the-art. Despite the simplicity of our pipeline, we only pay 6% on MSR-Action3D (96.9%, [Roz+16]). This is coherent with the fact that *intra-subject variability* is not totally absent in such a case ($\mathbf{p_{intra}} \approx 0.2$ in Table A.10), therefore mining the underlying assumption of our approach. Differently, we are scoring almost on par with respect to [Cav+16] (98.1%) on HDM-05, also improving the state-of-the-art on MSRC-Kinect12 by about 2% (95.0%, [Cav+16]).

Dataset	$\mathbf{p_{subject}}$	$\mathbf{p_{inter}}$	$\mathbf{p_{intra}}$	$\boldsymbol{\Delta}$
MSR-Action3D	0.78	0.86	0.19	0.71
MSRC-Kinect12	0.97	0.97	0.01	0.90
HDM-05	0.89	0.95	0.01	0.74

TABLE A.10: Quantitative evaluation of *inter* and *intra-subject variability*.

[5] To have a better insight of the importance of the knowledge of the subject who is performing the action, we have conducted an experiment on MSRC-Kinect12 using COV features where we assume that the correct *action-SVM* is not available. Using the best *action-SVMs* belonging to all other subjects the performance drops from 97.14% to 80.68% .

A.8 De-personalizing intention prediction: a leap between domain adaptation and action recogntion

The capacity of generalization is indeed a requested ability of any (action) recognition method. This is even more important in our case since the actual intention is never observed and such discriminants should be spot in very similar grasping actions for the different intentions. Indeed, in IfM, a better generalization can be implicitly achieved by identifying intention-specific subject-invariant discriminants which are embedded in the kinematics of a (apparently unrelated) grasping motor act. While these discriminants cannot easily be extracted, we can still take advantage of the bias with which each human subject is performing the initial grasping movement. To cope with this problem, in this Section we propose a new approach able to *explicitly* promote subject-independence for predicting intentions. This allows to actually improve cross-subject generalization and, consequently, performance.

Specifically, in Section A.8.1, we will show that a bias is actually present in the several subjects' action executions, and this makes one-subject-out validation extremely challenging. In Section A.8.2, we will present a novel technique, which is able to exploit these biases to improve the generalization capacity of the method, resulting in an ultimate superior performance for intention prediction. We will finally show that this approach also results profitable in standard action recognition benchmarks (MSR-Action3D [Li+10b] and HDM-05 [M+07]).

A.8.1 Multiple subjects, multiple biases

The accuracy results provided in the previous Sections are averaged across all the 17 subjects available in the dataset. A preliminary quantitative analysis to assess the bias among the subjects can be estimated by calculating the standard deviation (std) related to the average performance of the 3 best baseline methods above applied: std values result 13.40%, 14.14% and 16.12%, for ker-COV, DT-HOF-*VLAD* and AlexNet-OF-*VLAD* features, respectively.

As one can note, the standard deviation values are pretty high, meaning that accuracies are largely variable among the subjects. In other words, the generalization (subject independence) reached by the models on the new testing subject is not so high, which gives margin for improvement.

To further verify such claim, inspired by [Zun+17b], we performed another experiment in order to measure the bias provided by each subject. In [Zun+17b], leveraging on quantitative evidence of the high variances for the same action performed by different subjects, action recognition is formulated as a two-staged pipeline where, first, the subject is identified and second, its actions are recognized. Interestingly, for the task of subjects' identification, the same features exploited for discriminating actions are used, further denoting a clear evidence of the subject-related bias.

Inspired by such idea, in our case we used some of the baseline features that we previously presented in order to train a multi-class SVM to identify the 17 subjects in the proposed dataset. To do so, we adopted a one-intention-out

testing protocol where every trial referring to one single intention was left out for testing, while all the remaining trials were used for training. In Table A.11, we report the subjects' identification performance obtained after averaging across each intention left out. We register an outstanding performance of both ker-COV and DT-HOF-*VLAD* for subjects' identification, suggesting that intention prediction has a much stronger subject related bias with respect to classical action recognition problem. Differently, the performance of AlexNet-OF-*VLAD* is lower: presumably, after fine-tuning the network, a good intention prediction performance is already achieved by implicitly bridging the subjects-related biases.

ker-COV [3D]	DT-HOF-*VLAD* [2D]	AlexNet-OF-*VLAD* [2D]
97.25%	100%	53.34%

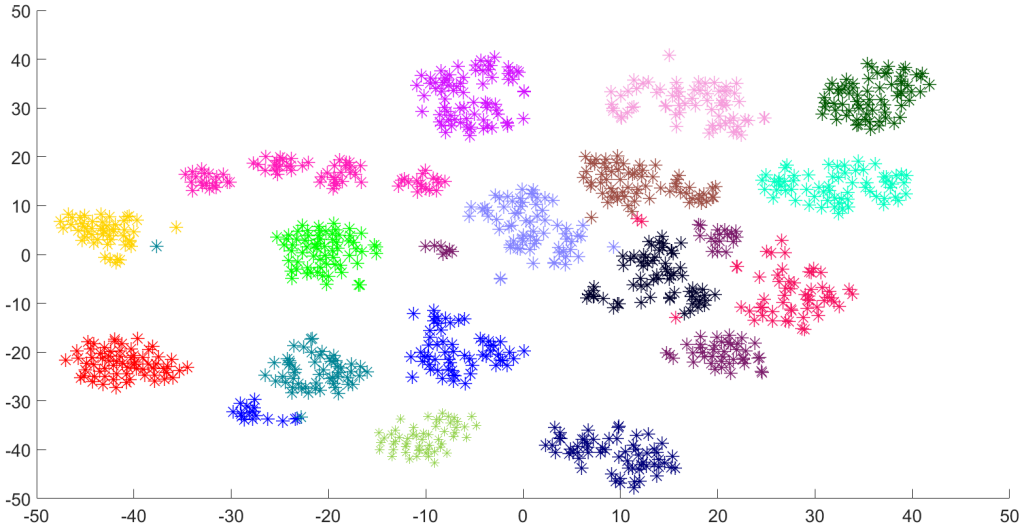
TABLE A.11: Subjects' identification performance.

The results in Table A.11 are also corroborated by the t-distributed Stochastic Neighbor Embedding (t-SNE) technique [MH08], the most used state-of-the-art visualization method. We applied it to ker-COV, DT-HOF-*VLAD* and AlexNet-OF-*VLAD*, obtaining the plots reported in Fig. A.8. Let us stress that t-SNE is a fully unsupervised method which does not exploit neither actions' nor intentions' labels. Nevertheless, ker-COV and DT-HOF-*VLAD* representations are perfectly able to cluster in 17 groups, each one corresponding to a single subject. The information of the subject who performed the grasping is clearly present in such representations and this can be seen as a bias which needs to be removed when training an intention predictor (see Fig. A.8(a) and (b)).
On the other hand, we are also able to explain why AlexNet-OF-*VLAD* features are not perfect in classifying the subject (see Fig. A.9(c)). In fact, the t-SNE plot (in Fig. A.9(d)) shows how, apparently, the fine-tuning process has achieved a nice separation of Placing intention (in cyan) vs. the others by mixing all the subjects.

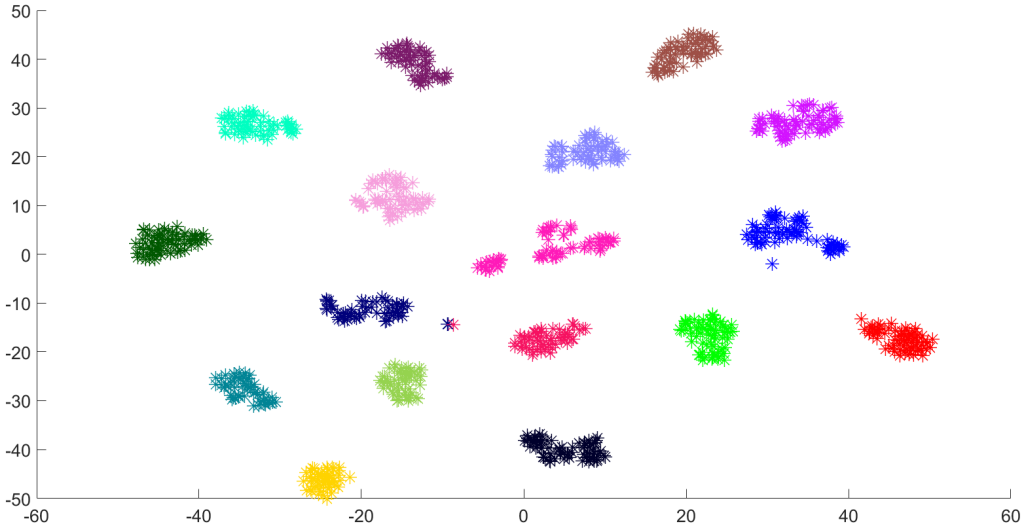
In summary, we have empirically proved the existence and the impact of subject related biases for intention prediction, being this trend more critical than in action recognition [Zun+17b]. Therefore, achieving intention prediction in a generalizable manner across subjects is difficult task, being nevertheless paramount for deploying an actual recognition system. In the following section, we will propose a novel approach to properly tackle this problem of generalizing across subjects and show that reducing this bias is beneficial for the sake of intentions' prediction.

A.8.2 Subject-Adversarial Domain Adaptation

To reduce the bias generated by the different agents, we resort to the idea to explicitly consider such information in devising a training method able to "confuse" the subjects such as to increase the generalization ability of the classification model. To this end, we propose a novel approach which is based on unsupervised domain adaptation [DM06]. This class of methods generically refers to a transfer learning problem where a model learnt from a certain amount of (source) data needs to be adapted on other (target) data, which is



(a) t-SNE on ker-COV features – each color represents one subject.



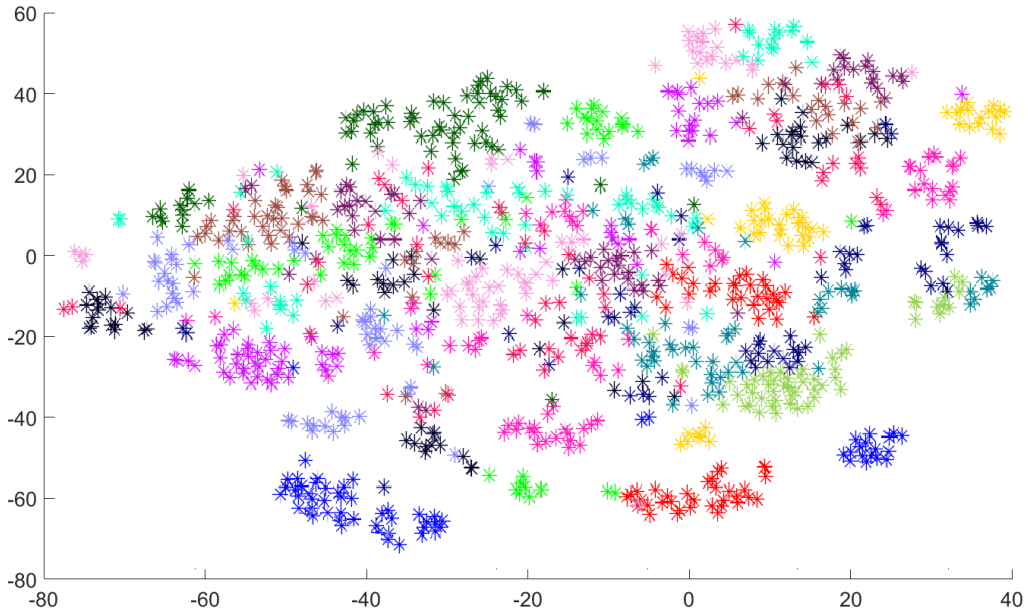
(b) t-SNE on DT-HOF-*VLAD* features – each color represents one subject.

FIGURE A.8: Bi-dimensional embedding of ker-COV, DT-HOF-*VLAD* and AlexNet-OF-*VLAD* using t-distributed Stochastic Neighbor Embedding. See also Figure A.9 Best viewed in color.

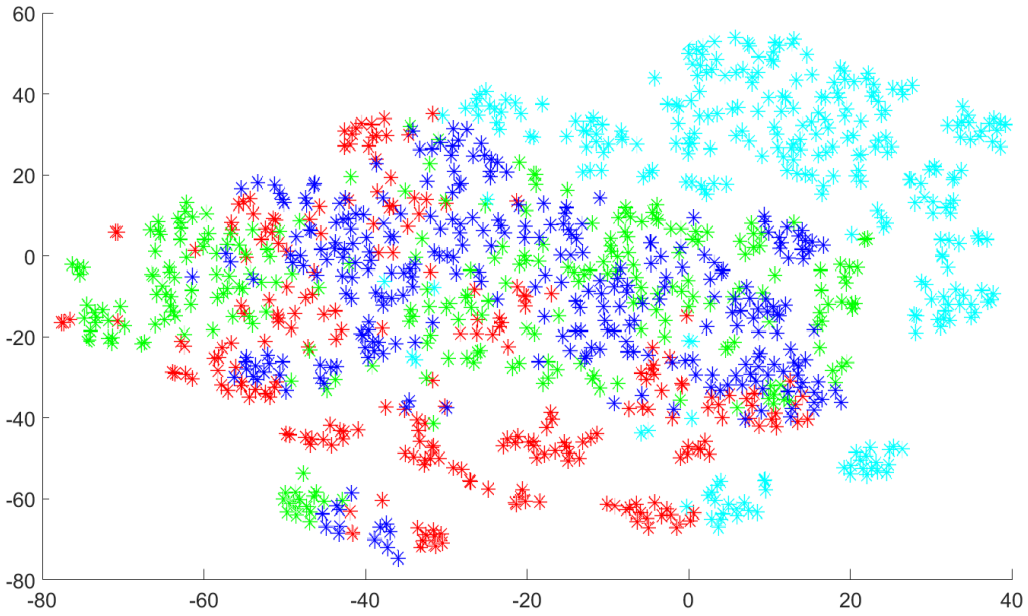
typically drawn from a different distribution, under some assumptions between source and target data domains.

Leveraging on the subjects' related biases discussed above, the intuition here is to consider each subject as a domain and subsequently perform adaptation. Therefore, we consider a multi-domain case where we exploit multiple (source) subjects in order to adapt our models to perform well on a new, unknown (target) agent.

We adopt adversarial domain adaptation for action recognition by learning a shared feature representation between subjects which can be effective for intention disambiguation. We want to learn a representation which, at the same time, leads to a top-scoring intention classifier and to a random chance scoring discriminator of the subjects [Gan+16]. This can be obtained employing adversarial training by means of the min-max formulation described in the



(a) t-SNE on AlexNet-OF-*VLAD* features – each color represents one subject.



(b) t-SNE on AlexNet-OF-*VLAD* features – each color represents one intention.

FIGURE A.9: Bi-dimensional embedding of ker-COV, DT-HOF-*VLAD* and AlexNet-OF-*VLAD* using t-distributed Stochastic Neighbor Embedding. Continues from figure A.8. Best viewed in color.

following.

We consider each grasping a bottle movement in our dataset $\mathcal{D}$ as a triplet $[\mathbf{x}, s, y]$, where $\mathbf{x}$ is a low-level representation for the grasping, s is the subject's label, and y is the intention's label (see Fig. A.10).

We look for a feature representation $\mathbf{f}(\mathbf{x}|\mathbf{W_f})$, depending on some parameters $\mathbf{W_f}$, which is trained to be intention-discriminative and subject-invariant. This

is achieved through the following optimization problem:

$$\min_{\mathbf{W_f},\mathbf{W}_i} \sum_{[\mathbf{x},s,y]\in\mathcal{D}} \ell_i(y, g(\mathbf{f}(\mathbf{x}|\mathbf{W_f}), \mathbf{W}_i)) \tag{A.4}$$

$$\min_{\mathbf{W}_s} \max_{\mathbf{W_f}} \sum_{[\mathbf{x},s,y]\in\mathcal{D}} \ell_s(s, h(\mathbf{f}(\mathbf{x}|\mathbf{W_f}), \mathbf{W}_s)). \tag{A.5}$$

The weights $\mathbf{W}_i$ in Eq. (A.4) and $\mathbf{W}_s$ in Eq. (A.5) are optimized in order to devise effective intention and subject related classifiers - g and h, respectively. However, g and h are both fed with the feature representation $\mathbf{f}$ which undergoes the following adversarial training. The learning on $\mathbf{f}$ is performed to promote, at the same time an efficient intention discrimination (ℓ_i is minimized with respect to $\mathbf{W_f}$) and a poor subject identification (ℓ_s is maximized with respect to $\mathbf{W_f}$)

As far as we know, (adversarial) domain adaptation has never been applied to neither action recognition nor its variants (Fig. A.1). In this work, we are using that for the first time by considering the following two settings, demonstrating that it is indeed suitable for intention prediction.

Subject-Adversarial Domain Adaptation (SADA). The SADA approach is derived from the unsupervised domain adaptation pipeline where the unannotated target data (here, the testing subject) is used to modify the feature representation, while the learning phase of the classifier for the main task is done on the source domains only (here, the training subjects' actions). In practice, the source domain data is used to learn the classifier to discriminate actions (intentions in our case) in a supervised way, whereas the target domain data is still used in training, but in an unsupervised way, since action labels are unknown (we only use the information that the test subject's identity is different from that of any other training subjects). In our case, the actions of the test subject are our target domain while the actions of all the other subjects constitute the source domain: we aim at training the system by improving the action classification performance while minimizing the capability of the system to identify the subject who executed that action.

Blind-SADA. Blind-SADA can be seen as a generalization of the classical domain adaptation setting which, overall, relies on the fact that the target domain is fixed and specified. In fact, even in the unsupervised case, unannotated target data are exploited during learning to adapt with the source. Here, differently, we posit that the availability of multiple source domains (*i.e.* , training subjects) can provide enough information as to learn an adaptation which is enough powerful to be *blindly* applied to an arbitrary target domain (*i.e.* , testing subjects), without exploiting target data in any way during the learning stage.
We also explored this setting since in a general video-surveillance framework, a system should be able to perform well on a variety of unknown, never seen, testing subjects, still ensuring a high generalization in predicting humans' intention. In this situation, it is desirable to investigate whether subjects' confusion (A.5) applied on a fixed number of subjects is still generalizable to other, unseen ones. In our experiments, we do this by optimizing Eqs. (A.4) and (A.5) by only using the data of 16 subjects, without using the data of the test subject left out neither for subjects' confusion (differently from SADA),

Data type	Method	baseline	Blind-SADA	SADA
3D	ker-COV	71.57	73.13 ($\lambda = 0.6$)	**80.48** ($\lambda = 0.1$)
2D	DT-HOF-*VLAD*	56.01	57.15 ($\lambda = 1.5$)	**70.42** ($\lambda = 0.2$)
2D	AlexNet-OF-*VLAD*	65.64	66.59 ($\lambda = 1$)	**67.95** ($\lambda = 0.1$)

TABLE A.12: Subject adaptation results on IfM. In brackets the best setting of λ.

nor for tuning the parameter of the intention prediction branch (A.4) (see Fig. A.10).

Connections with privileged information. We can interpret Blind-SADA within the paradigm of privileged information [LP+15], which stands for training with additional information, the latter being not available in testing. In our case, such additional information stands for the subjects' labels: when Blind-SADA tries to adapt for multiple subjects as to generalize to a generic (and unknown) subject, we definitely use the training subjects' identity to perform adaptation, while the identity of the testing subject is actually unknown.

Implementation

Technically, SADA and its Blind variant are implemented by the architecture inspired by [Gan+16] and named *Subject-Adversarial Neural Network* (SANN), which is composed by 3 modules. A first, low-level, network module learns the feature representation $\mathbf{f}(\mathbf{x}|\mathbf{W_f})$ - blue in Fig. A.10. After that, two separate intention- (green in Fig. A.10) and subject-related modules (yellow in Fig. A.10) are responsible for the intention-discrimination and subjects-confusion, respectively. As previously explained (Section A.8.2), the blue module is responsible for achieving a representation which, at the same time optimizes the green module (intentions) and fool the yellow one (subjects). Therefore, our adversarial approach stands from the fact that the yellow module seeks for a perfect subjects' discrimination built on top of a feature representation which is learnt to be subject-invariant in the blue module.

SANN is trained accordingly to the one-subject-out protocol adopted in this work. It is important to note that, for all-class comparison considered, we train one SANN per subject left out for testing, using the remaining subjects as multiple source domains. Performance of each network are evaluated on the subject left out and results are averaged across.

More specifically, in SADA, the subject confusion module is fed with the unlabeled (as for the intention) data of the testing subjects to adapt the feature $\mathbf{f}(\mathbf{x}|\mathbf{W_f})$ to the specific agent. Differently, Blind-SADA *never* exploits the trials of the testing subject in training and performs adaptation by totally ignoring the target domain (both identities and intention labels).

We accommodate the publicly available code[6] of [Gan+16] to deal with a different number of subjects to perform adaptation. Indeed, [Gan+16] considers a simplified setting of one target domain only, whereas, differently, we consider multiple domains. The optimization of (A.4) and (A.5) is carried out by using a joint back-propagation In particular, we compute the updates on the parameters $\mathbf{W}_s$ and $\mathbf{W}_i$ separately on the two branches. Then, we used the gradient

[6] http://graal.ift.ulaval.ca/dann/

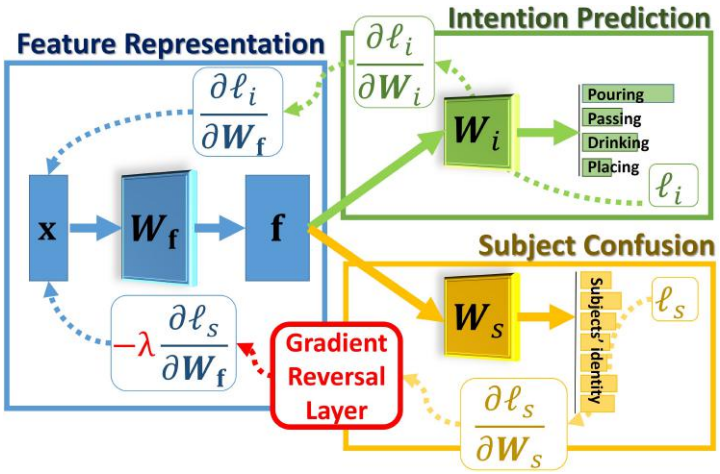


FIGURE A.10: The adopted Subject-Adversarial Neural Network (SANN). Best viewed in color.

reversal layer [Gan+16] to change the sign of the derivative of the subject loss ℓ_s with respect to $\mathbf{W_f}$ (after a re-scaling by a parameter λ). The derivative of ℓ_i with respect to $\mathbf{W_f}$ is instead back-propagated with the correct sign (see Fig. A.10).
A multi-layer perceptron (MLP) network with one hidden layer of dimension 500 was designed as the shared feature representation $\mathbf{f}(\mathbf{x}|\mathbf{W_f})$, where, as $\mathbf{x}$, we considered either ker-COV, DT-HOF-*VLAD* or AlexNet-OF-*VLAD* features. For the intention prediction module, we trained a four-way softmax function using a cross entropy loss for ℓ_i. Similarly, for the subject confusion module, a 17- or 16-way cross-entropy loss is used for ℓ_s in SADA and Blind-SADA, respectively.

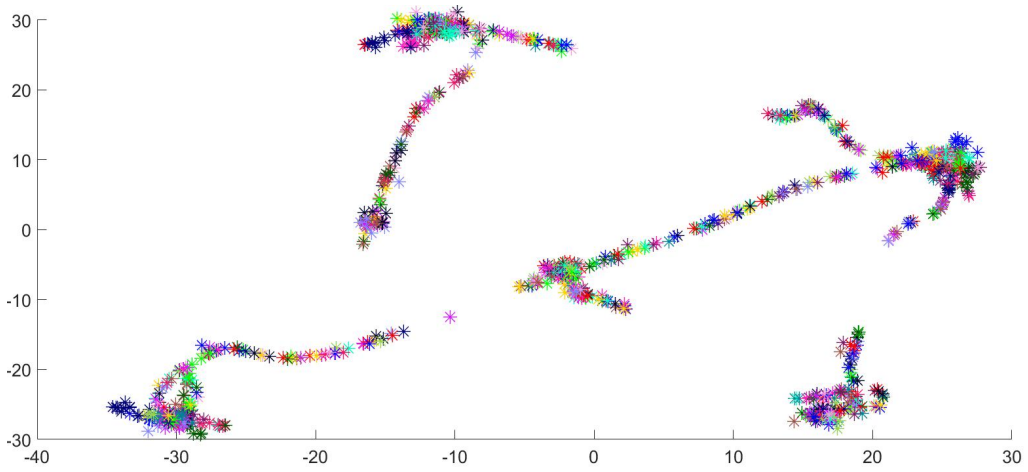
We cross-validate λ by selecting the value which maximally fool the subjects' classifier in the subject confusion module.

As a common pre-processing step on data, we run PCA on the ker-COV, DT-HOT-*VLAD* and AlexNet-OF-*VLAD*, retaining the 99.5% of explained variance: this step is only required to speed up the computation and we did not register a major effect on performance.

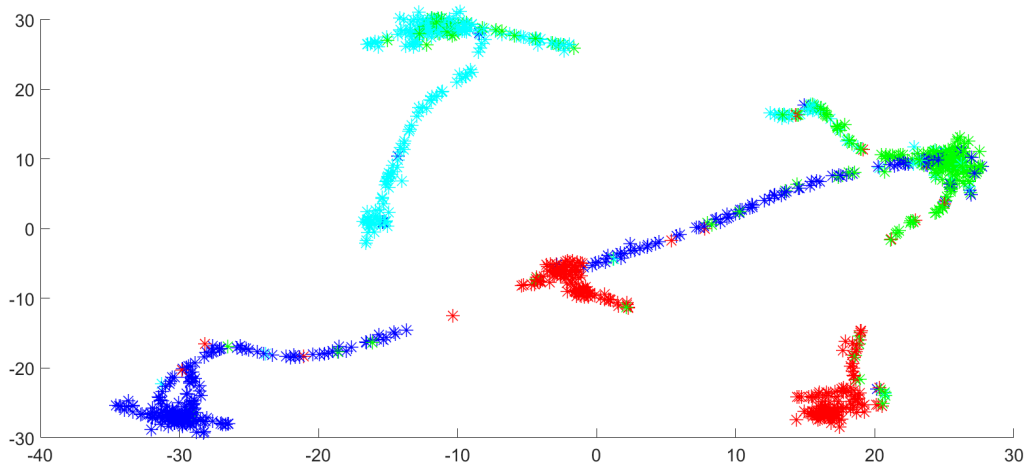
Results and discussion

In Table A.12, we report the results corresponding to SADA and Blind-SADA, as compared with a baseline method. The baseline is a simple MLP neural network composed by the feature representation and the intention prediction modules of SANN only (blue+green boxes in Fig. A.10), without applying any subject confusion.

Baseline performance are almost the same as those presented in Tables A.1, A.2 and A.3, where we used an SVM. In both MLP and SVM techniques, only the intention label information is exploited and the final performance is comparable (*e.g.* , 58.23% in Table A.2 vs. 56.01% in Table A.12 for DT-HOF-*VLAD*).
In the last column of Table A.12, we note instead a large improvement using SADA in the main intention prediction task: +8.91% for ker-COV, +14.41%



(a) t-SNE using ker-COV features and Blind-SADA – each color represents one subject.



(b) t-SNE using ker-COV features and Blind-SADA – each color represents one intention.

FIGURE A.11: Bi-dimensional embedding using t-Distributed Stochastic Neighbor Embedding for the ker-COV features transformed with Blind-SADA. Best viewed in color.

for DT-HOF-*VLAD*, and +2.31% for AlexNet-OF-*VLAD*. The largest improvements are obtained considering DT-HOF-*VLAD* for video data and the 3D-based ker-COV encoding. These two neatly increased scores support the t-SNE visualization in Fig. A.8(a) and A.8(b), showing almost perfect compact clusters per subject with ker-COV or DT-HOF-*VLAD* features. This framework proved to be able to remove the predominant subject information from the data samples and to get better performance in the multi-class intention prediction task.

The CNN features deserve a separate discussion since the improvement with domain adversarial training is not huge although still present. We guess that the fine tuning process operated for CNN feature extraction already reduces the impact of the subject-related biases to some extent. In other words, CNN fine tuning already performs a sort of domain adaptation and subject confusion (as visible in t-SNE plots in Fig. A.9(a) and A.9(b)), hence our framework is less effective in this case.

The results of Blind-SADA are reported in the third column of Table A.12. The improvement with respect to the baseline approach is smaller than SADA, but

still significant: +1.56% for ker-COV, +1.14% for DT-HOF-*VLAD*, and +0.95% AlexNet-OF-*VLAD*. Hence, we can still assert that training the net with the proposed SANN framework is effective for intention prediction. This means that, also relaxing the classic domain adaptation framework, subject confusion is also beneficial when the target domain is not utilized during training, since a hidden representations could still be learnt to discriminate better between the intentions, reducing the noisy knowledge (*i.e.* , the bias) coming from the subject identities.
To get a deeper insight on how the features are transformed by means of Blind-SADA training process, we plot in Fig. A.11 the hidden representation of ker-COV when one subject trials are left fully out (in this case, subject 1). If we compare the t-SNE representations in Fig. A.8(a) with those in Figs. A.12(a) and A.12(b), we can note that the new ker-COV hidden representations are no more grouped in compact clusters associated to subjects. In Fig. A.12(a), the subjects are totally mixed whereas the samples are rearranged better for the main intention prediction task, as visible in Fig. A.12(b). This suggests that the training process has still learned feature discriminants for the intentions, at the expense of making indistinguishable the subjects, which was exactly our goal. Actually, if now we try to perform the subjects' identification experiment over the hidden representation plotted in Fig. A.12(a) and A.12(b), the average accuracy drops from 97.25% (Table A.11) to 6.88% coherently obtaining an almost random chance performance in subject identification.

SADA on action recognition datasets

Grounding on the experimental findings discovered in IfM, we aim now to assess the SADA framework in public action recognition datasets to see if we can gain in accuracy performance. The necessary condition in which we can apply the proposed pipeline is having access to action and subjects labels at the same time for each trial.

For this purpose, we considered the public MSR-Action3D [Li+10b] and HDM-05 [M̈+07], using the off-the-shelf covariance feature (COV) representation utilized in [Zun+17b].

	baseline	Blind-SADA	SADA
MSR-Action3D	80.41	81.73 ($\lambda = 0.1$)	**84.84** ($\lambda = 0.6$)
HDM-05	94.68	95.41 ($\lambda = 0.2$)	**95.93** ($\lambda = 0.4$)

Table A.13: Subject adaptation on MSR-Action3D and HDM-05 action recognition benchmarks. In brackets, the value of λ adopted.

As done for IfM, we carried out a one-subject-out testing procedure where each subject is left out for testing, and we fed COV features into the baseline, SADA and Blind-SADA architectures. As previously done, λ is chosen by cross-validation by selecting the value which best achieves subject confusion. The results in Table A.13 show a trend similar to that of Table A.12, where performance improves from the baseline passing to the Blind-SADA, finally registering the largest improvement of +4.43% in MSR-Action3D and +1.25% in HDM-05 using the SADA framework. Therefore, on these datasets we

retrieve the same findings of IfM, certifying that the same approach can also be beneficial for the classic action recognition problem.

A.9 Conclusions

In this Appendix, we propose Intention from Motion, a novel problem of predicting the goal which originates from an human action by using the kinematics only, in a context-free setting. We present a new dataset and find that by only inspecting grasping-a-bottle actions, we can predict whether they fulfill a Pouring, Passing, Drinking or Placing intention.

As the result of a broad baseline analysis, we prove that our novel problem is feasible and intention discriminants are embedded in the anticipative and apparently unrelated grasping motor act.

We prove that the personalized approach to spot intentions is effective in overcoming the biases that are related to each single subject. In addition to the sound performance in intention prediction, we also obtained solid results on action recognition benchmarks, showing that the personalized perspective is a very effective way of ensuring high recognition performance by customizing the classifier on the specific human agent whose actions need to be recognized or its intentions predicted.

However, since the latter approach is detrimental for the generalization capability of the intention prediction system, we propose an opposite approach to personalization. Namely we pursue a de-personalization stage by casting action recognition and its variants as domain adaptation. When interpreting each subject as a domain, Subject-Adversarial Domain Adaptation (SADA) remarkably boosts the prediction capability for intentions when the test subjects are pre-determined (and its unannotated trials are exploited to guide the adaptation of the feature representation).

As an extension, we propose Blind-SADA and show that exploiting subject's identities in training to perform adaptation leads to good generalization on an unknown agent. Despite less data are exploited by Blind-SADA, its performance is not too far degraded from the one of SADA, and both improve upon the baseline. This certifies the effectiveness of our idea of learning from multiple subjects as to adapt on both specific and general target domains/subjects.

Future directions of this work aim at the design of intention prediction systems in more real-world scenarios and to perform a fine-grained analysis to extract interpretable representations of the motion patterns as well as to locate in space/time the intention discriminants embedded in the kinematics.

Bibliography

[Abn02] S. Abney. “Bootstrapping”. In: *Annual Meeting of the Association for Computational Linguistics*. 2002.

[Aho+06] T. Ahonen, A. Hadid, and M. Pietikainen. “Face Description with Local Binary Patterns: Application to Face Recognition”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 28.12 (2006), pp. 2037–2041.

[An+07] S. An, W. Liu, and S. Venkatesh. “Face Recognition Using Kernel Ridge Regression.” In: *IEEE Computer Vision and Pattern Recognition - CVPR*. 2007.

[Ani+15] R. Anirudh, P. Turaga, J. Su, and A. Srivastava. “Elastic Functional Coding of Human Actions: From Vector-Fields to Latent Variables”. In: *IEEE Computer Vision and Pattern Recognition - CVPR*. 2015.

[Ans+14] C. Ansuini, A. Cavallo, C. Bertone, and C. Becchio. “Intentions in the Brain: The Unveiling of Mister Hyde”. In: *The Neuroscientist* 21 (2014), pp. 126–135.

[Ans+15] C. Ansuini, A. Cavallo, A. Koul, M. Jacono, Y. Yang, and C. Becchio. “Predicting object size from hand kinematics: a temporal perspective”. In: *PloS One* 2.10 (2015).

[AR11] J. K. Aggarwal and M. S. Ryoo. “Human Activity Analysis: A Review”. In: *ACM Computing Surveys* 43.3 (2011).

[Arg+08] A. Argyriou, T. Evgeniou, and M. Pontil. “Convex multi-task feature learning”. In: 73.3 (2008), pp. 243–272.

[Ars+06] V. Arsigny, P. Fillard, X. Pennec, and N. Ayache. In: *SIAM Journal on Matrix Analysis and Applications* 29.1 (2006), pp. 328–347.

[Ars+07] V. Arsigny, P. Fillard, X. Pennec, and N. Ayache. “Geometric Means in a Novel Vector Space Structure on Symmetric Positive-Definite Matrices”. In: *SIAM J. Matrix Anal. Appl.* 29.1 (2007), pp. 328–347.

[Atr+10] P. K. Atrey, M. A. Hossain, A. El Saddik, and M. S. Kankanhalli. “Multimodal fusion for multimedia analysis: a survey”. In: *Multimedia Systems* 16 (2010), pp. 345–379.

[AZ05] R. K. Ando and T. Zhang. “A Framework for Learning Predictive Structures from Multiple Tasks and Unlabeled Data”. In: 6 (2005), pp. 1817–1853.

[Bal+04] M. Balcan, A. Blum, and Y. Ke. “Co-training and expansion: Towards bridging theory and practice”. In: *UCLA Computer Science Department Technical Reports*. 2004.

[Bay+13] J. Bayer, C. Osendorfer, and N. Chen. “On Fast Dropout and its Applicability to Recurrent Networks”. In: *arXiv:1311.0701*. 2013.

[Bel+06] M. Belkin, P. Niyogi, and V. Sindhwani. "Manifold Regularization: A Geometric Framework for Learning from Labeled and Unlabeled Examples". In: 7 (2006), pp. 2399–2434.

[Ben+09] Y. Bengio, J. Louradour, R. Collobert, and J. Weston. "Curriculum Learning". In: *International Conference on Machine Learning - ICML*. 2009.

[Bha15] R. Bhatia. *Positive Definite Matrices*. Princeton, NJ, USA: Princeton University Press, 2015.

[Bia+13] M. S. Biagio, M. Crocco, M. Cristani, S. Martelli, and V. Murino. "Heterogeneous Auto-similarities of Characteristics (HASC): Exploiting Relational Information for Classification". In: *IEEE International Conference on Computer Vision - ICCV*. 2013.

[Bis06] C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.

[Bis95] C. M. Bishop. "Training with Noise is Equivalent to Tikhonov Regularization". In: *Neural Computation* 7.1 (1995), pp. 108–116.

[Blo+12] V. Bloom, D. Makris, and V. Argyriou. "G3D: A Gaming Action Dataset and Real Time Action Recognition Evaluation Framework". In: *IEEE Computer Vision and Pattern Recognition - CVPR*. 2012.

[BM98] A. Blum and T. Mitchell. "Combining labeled and unlabeled data with co-training". In: *ACM Conference on Learning Theory - COLT*. 1998.

[Boi+08] O. Boiman, E. Shechtman, and M. Irani. "In defense of Nearest-Neighbor based image classification". In: *IEEE Computer Vision and Pattern Recognition - CVPR*. 2008.

[Boo+16] L. Boominathan, S. S. Kruthiventi, and R. V. Babu. "Crowdnet: a deep convolutional network for dense crowd counting". In: *ACM Conference on Multimedia - ACMM*. 2016.

[Bou+16] K. Bousmalis, G. Trigeorgis, N. Silberman, D. Krishnan, and D. Erhan. "Domain Separation Networks". In: *Advances on Neural Information and Processing Systems - NIPS*. 2016.

[Boy+11] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. "Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers". In: 3.1 (2011), pp. 1–122.

[Bro+12] G. Brown, A. Pocock, M.-J. Zhao, and M. Lujan. "Conditional Likelihood Maximization: A unifying framework for information theoretic feature selection". In: 13 (2012), pp. 27–66.

[BS04] S. Bickel and T. Scheffer. "Multi-view clustering." In: *International Conference on Data Mining - ICDM*. 2004.

[BS13] P. Baldi and P. Sadowski. "Understanding Dropout". In: *Advances in Neural Information Processing Systems - NIPS*. 2013.

[BS14] P. Baldi and P. Sadowski. "The Dropout Learning Algorithm". In: *Artificial Intelligence*. 2014.

[Bub+13] D. N. Bub, M. E. J. Masson, and T. Lin. "Features of Planned Hand Actions Influence Identification of Graspable Objects". In: *Psychological Science* 24.7 (2013), pp. 1269–1276.

[BV04] S. Boyd and L. Vandenberghe. *Convex Optimization*. New York, NY, USA: Cambridge University Press, 2004. ISBN: 0521833787.

[Cab+11] R. S. Cabral, F. Torre, J. P. Costeira, and A. Bernardino. "Matrix completion for multi-label image classification". In: *Advances on Neural Information and Processing Systems - NIPS*. 2011.

[Cag+17] G. Caglar, S. Jose, M. Marcin, and Y. Bengio. "A Robust Adaptive Stochastic Gradient Method for Deep Learning". In: *CoRR:1703.00788*. 2017.

[Cai+10] J.-F. Cai, E. J. Candes, and Z. Shen. "A singular value thresholding algorithm for matrix completion". In: *SIAM Journal on Optimization* 4.20 (2010), 1956–1982.

[Can86] J Canny. "A Computational Approach to Edge Detection". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 8.6 (1986), pp. 679–698.

[Cao+13] Y. Cao, D. Barrett, A. Barbu, N. Siddharth, H. Yu, A. Michaux, Y. Lin, S. Dickinson, J. M. Siskind, and S Wang. "Recognizing Human Activities from Partially Observed Videos". In: *IEEE Computer Vision and Pattern Recognition - CVPR*. 2013.

[Car+06] C. Carmeli, E. De Vito, and A. Toigo. "Vector valued reproducing kernel Hilbert spaces of integrable functions and Mercer theorem". In: *Journal of Applied Analysis* 4 (4 2006), pp. 377 –408.

[Car+11] I. Carpinella, J. Jonsdottir, and M. Ferrarin. "Multi-finger coordination in healthy subjects and stroke patients: a mathematic modelling approach". In: *Journal of Neuroengineering Rehabilitation* 8.1 (19 2011).

[Car+17] F. M. Carlucci, L. Porzi, B. Caputo, E. Ricci, and S. Rota Bulo. "AutoDIAL: Automatic DomaIn Alignment Layers". In: *IEEE International Conference on Computer Vision - ICCV*. 2017.

[Cav+16] J. Cavazza, A. Zunino, M. San Biagio, and V. Murino. "Kernelized covariance for action recognition". In: *IAPR International Conference on Pattern Recognition - ICPR*. 2016.

[Cav+17a] J. Cavazza, P. Morerio, and V. Murino. "A Compact Kernel Approximation for 3D Action Recognition". In: *International Conference on Image Analysis and Processing - ICIAP*. 2017.

[Cav+17b] J. Cavazza, P. Morerio, B. Haeffele, C. Lane, V. Murino, and R. Vidal. "Dropout as Low-Rank Regularizer for Matrix Factorization". In: *arXiv*. 2017.

[Cav+17c] J. Cavazza, P. Morerio, and V. Murino. "When Kernel Methods meet Feature Learning: Log-Covariance Network for Action Recognition". In: *IEEE Computer Vision and Pattern Recognition - CVPR - workshops*. 2017.

[CC14] K. Cho and X. Chen. "Classifying and Visualizing Motion Capture Sequences using Deep Neural Networks". In: *CoRR:1306.3874*. 2014.

[Cha+08] A. B. Chan, J. S. Zhang, and L. N. Vasconcelos. "Privacy preserving crowd monitoring: Counting people without people models or tracking". In: *IEEE Computer Vision and Pattern Recognition - CVPR* (2008).

[Cha+09a] A. Chan, M Morrow, and N. Vasconcelos. "Analysis of crowded scenes using holistic properties". In: *IEEE Computer Vision and Pattern Recognition - CVPR- workshops*. 2009.

[Cha+09b] R. Chaudhry, A. Ravichandran, G. Hager, and R. Vidal. "Histograms of oriented optical flow and Binet-Cauchy kernels on

nonlinear dynamical systems for the recognition of human actions". In: *IEEE Computer Vision and Pattern Recognition - CVPR*. 2009.

[Cha+13] R. Chaudhry, F. Ofli, G. Kurillo, R. Bajcsy, and R. Vidal. "Bioinspired dynamic 3d discriminative skeletal features for human action recognition". In: *IEEE Computer Vision and Pattern Recognition - CVPR- workshops*. 2013.

[Che+12a] K. Chen, C. C. Loy, S. Gong, and T. Xiang. "Feature Mining for Localised Crowd Counting". In: *British Machine Vision Conference - BMVC*. 2012.

[Che+12b] A. Cherian, S. Sra, A. Banerjee, and N. Papanikolopoulos. "Jensen-Bregman LogDet Divergence with Application to Efficient Similarity Search for Covariance Matrices". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35.9 (2012), pp. 2161–2174.

[Che+13a] K. Chen, S. Gong, T. Xiang, and C. Change Loy. "Cumulative Attribute Space for Age and Crowd Density Estimation". In: *CVPR*. 2013.

[Che+13b] A. Cherian, S. Sra, A. Banerjee, and N. Papanikolopoulos. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35.9 (2013), pp. 2161–2174.

[Chi+16] W.-L. Chiang, M.-C. Lee, and C.-J. Lin. "Parallel Dual Coordinate Descent Method for Large-scale Linear Classification in Multicore Environments". In: *International Conference on Knowledge Discovery and Data Mining - KDD*. 2016.

[Ché+15] G. Chéron, I. Laptev, and C. Schmid. "P-CNN: Pose-based CNN Features for Action Recognition". In: *IEEE International Conference on Computer Vision - ICCV*. 2015.

[CL10] K. Crammer and D. Lee. "Learning via Gaussian Herding". In: *Advances on Neural Information and Processing Systems - NIPS*. 2010.

[CM15] J. Cavazza and V. Murino. "People Counting by Huber Loss Regression". In: *IEEE International Conference on Computer Vision - ICCV - workshops*. 2015.

[CM16] J. Cavazza and V. Murino. "Active Regression with Adaptive Huber Loss". In: *arXiv:1606.01568* (2016).

[CP12] A. Conversano and A. Pillay. "Connected components of definable groups and -minimality I". In: *Advances in Mathematics* 231.2 (2012), pp. 605 –623. ISSN: 0001-8708. DOI: https://doi.org/10.1016/j.aim.2012.05.022. URL: http://www.sciencedirect.com/science/article/pii/S0001870812002095.

[CR09] E. J. Candès and B. Recht. "Exact Matrix Completion via Convex Optimization". In: *Foundations of Computational Mathematics* 9.6 (2009), p. 717.

[Cra+09] K. Crammer, A. Kulesza, and M. Dredze. "Adaptive Regularization of Weight Vectors". In: *Advances on Neural Information and Processing Systems - NIPS*. 2009.

[CRC14] A. Chakraborty and K. Roy-Chowdhury. "Context-Aware Activity Forecasting". In: *Asian Conference on Computer Vision*. 2014.

[CT10] E. J. Candès and T. Tao. "The power of convex relaxation: Near-optimal matrix completion". In: *IEEE Transactions on Information Theory* 56.5 (2010), pp. 2053–2080.

[CV12] A. B. Chan and N. Vasconcelos. "Counting people with low-level features and bayesian regression". In: *IEEE Transactions on Image Processing - TIP* 21.4 (2012), pp. 2160–2177.

[Dal+06] N. Dalal, B. Triggs, and C. Schmid. "Human Detection Using Oriented Histograms of Flow and Appearance". In: *European Conference on Computer Vision*. 2006.

[Das+01] S. Dasgupta, M. L. Littman, and D. A. McAllester. "PAC Generalization Bounds for Co-training". In: *Advances on Neural Information and Processing Systems - NIPS*. 2001.

[Dav+95] A. C. Davies, J. H. Yin, and S. A. Velastin. "Crowd monitoring using image processing". In: *Electronics & Communication Engineering Journal* 7 (1995), pp. 37–47.

[Den+09] J. Deng, W. Dong, R. Socher, L. jia Li, K. Li, and L. Fei-fei. "Imagenet: A large-scale hierarchical image database". In: *IEEE Computer Vision and Pattern Recognition - CVPR*. 2009.

[Den+14] Z. H. Deng, K. S. Choi, and S. Jiang Y. Z. Wang. "Generalized hidden-mapping ridge regression, knowledge-leveraged inductive transfer learning for neural networks, fuzzy systems and kernel methods". In: *IEEE Transactions on Cybernetics*. 44.12 (2014), 2585–2599.

[Dev+14] M. Devanne, S. Wannous, S. Berretti, P. Pala, P. Daoudi, and A. Del Bimbo. "3D Human Action Recognition by Shape Analysis of Motion Trajectories on Riemannian Manifold". In: *IEEE Transactions on Cybernetics*. 2014.

[DM06] H. Daumé III and D. Marcu. "Domain Adaptation for Statistical Classifiers". In: *Elsevier Journal of Artificial Intelligence Research - JAIR* 26.1 (2006), pp. 101–126.

[Don+14] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell. "DeCAF: A Deep Convolutional Activation Feature for Generic Visual Recognition". In: *International Conference on Machine Learning - ICML*. 2014.

[Don+16] A. Dong, F.-l. Chung, Z. Deng, and S. Wang. "Semi-Supervised SVM With Extended Hidden Features". In: *IEEE Transactions on Cybernetics* 46.12 (2016), pp. 2924–2937.

[Dry+09] I. L. Dryden, A. Koloydenko, and D. Zhou. "Non-Euclidean statistics for covariance matrices, with applications to diffusion tensor imaging". In: *The Annals of Applied Statistics* 3.3 (2009), pp. 1102–1123.

[DT05] N. Dalal and B. Triggs. "Histograms of oriented gradients for human detection". In: *IEEE Computer Vision and Pattern Recognition - CVPR*. 2005.

[DT06] J. W. Davis and A. Tyagi. "Minimal-latency human action recognition using reliable-inference." In: *Image and Vision Computing* 24.5 (2006), pp. 455–472.

[Du+15] Y. Du, W. Wang, and L. Wang. "Hierarchical Recurrent Neural Network for Skeleton Based Action Recognition". In: *IEEE Computer Vision and Pattern Recognition - CVPR*. 2015.

[Elk14] M. van Elk. "The left IPL represents stored hand-postures for object use and action prediction". In: *Frontiers in Psychology* 5.333 (2014).

[Eva+14] G. Evangelidis, G. Singh, and R. Horaud. "Skeletal quads: Human action recognition using joint quadruples". In: *IAPR International Conference on Pattern Recogntion - ICPR*. 2014.

[Evg+00] T. Evgeniou, M. Pontil, and T. Poggio. "Regularization Networks and Support Vector Machines". In: *Advances in Computational Mathematics* 13.1 (2000).

[Fer+13] B. Fernando, A. Habrard, M. Sebban, and T. Tuytelaars. "Unsupervised Visual Domain Adaptation Using Subspace Alignment". In: *IEEE International Conference on Computer Vision - ICCV*. 2013.

[FF+04] L. Fei-Fei, R. Fergus, and P. Perona. "Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories". In: *IEEE Computer Vision and Pattern Recognition - CVPR- workshops*. 2004.

[Fot+12] S. Fothergill, H. M. Mentis, P. Kohli, and S. Nowozin. "Instructing people for training gestural interactive systems". In: *ACM Conference on Multimedia - ACMM*. 2012.

[FZ14] D. F. Fouhey and C. Zitnick. "Predicting Object Dynamics in Scenes". In: *IEEE Computer Vision and Pattern Recognition - CVPR*. 2014.

[Gan+16] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky. "Domain-adversarial training of neural networks". In: 17.59 (2016), pp. 1–35.

[GB04] Y. Grandvalet and Y. Bengio. "Semi-supervised Learning by Entropy Minimization". In: *Advances on Neural Information and Processing Systems - NIPS*. 2004.

[GB08] M. Grant and S. Boyd. "Graph implementations for nonsmooth convex programs". In: *Recent Advances in Learning and Control*. Ed. by V. Blondel, S. Boyd, and H. Kimura. Lecture Notes in Control and Information Sciences. Springer-Verlag Limited, 2008, pp. 95–110.

[Ges+01] T. V. Gestel, J. A. K. Suykens, B. D. Moor, and J. Vandewalle. "Automatic relevance determination for Least Squares Support Vector Machines classifiers." In: *European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning - ESANN*. 2001.

[GG16] Y. Gal and Z. Ghahramani. "Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning". In: *International Conference on Machine Learning - ICML*. 2016.

[GL11] R. Gopalan and R. Li. "Domain adaptation for object recognition: An unsupervised approach". In: *IEEE International Conference on Computer Vision - ICCV*. 2011.

[GL15] Y. Ganin and V. S. Lempitsky. "Unsupervised Domain Adaptation by Backpropagation". In: *International Conference on Machine Learning - ICML*. 2015.

[Glo+11] X. Glorot, A. Bordes, and Y. Bengio. "Domain adaptation for large-scale sentiment classification: A deep learning approach". In: *International Conference on Machine Learning - ICML*. 2011.

[Gon+12] B. Gong, Y. Shi, F. Sha, and K. Grauman. "Geodesic flow kernel for unsupervised domain adaptation". In: *IEEE Computer Vision and Pattern Recognition - CVPR*. 2012.

[Goo+16] I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. MIT Press, 2016.

[Gri+07] G. Griffin, A. Holub, and P Perona. "Caltech-256 object category dataset". In: *Technical Report 7694, California Institute of Technology*. 2007.

[Gri12] A. Griewank. "Who invented the Reverse Mode of Differentiation?" In: *Optimization Stories* Documenta Matematica, Extra Volume ISMP (2012) (2012), pp. 389–400.

[Gud+08] S. Gudmundsson, T. P. Runarsson, and S. Sigurdsson. "Support Vector Machines and Dynamic Time Warping for Time Series". In: *International Joint Conference on Neural Networks - IJCNN*. 2008.

[Gup+14] S. Gupta, R. Girshick, P. Arbelaez, and J. Malik. "Learning rich features from RGB-D images for object detection and segmentation". In: *European Conference on Computer Vision*. 2014.

[Hae+14] B. D. Haeffele, E. Young, and R. Vidal. "Structured Low-Rank Matrix Factorization: Optimality, Algorithm, and Applications to Image Processing". In: *International Conference on Machine Learning - ICML*. 2014.

[Hae+17a] B. D. Haeffele, E. Young, and R. Vidal. "Global Optimality in Neural Network Training". In: *IEEE Computer Vision and Pattern Recognition - CVPR*. 2017.

[Hae+17b] P. Haeusser, T. Frerix, A. Mordvintsev, and D. Cremers. "Associative Domain Adaptation". In: *IEEE International Conference on Computer Vision - ICCV*. 2017.

[Hae+17c] P. Haeusser, A. Mordvintsev, and D. Cremers. "Learning by Association - A versatile semi-supervised training method for neural networks". In: *IEEE Computer Vision and Pattern Recognition - CVPR*. 2017.

[Hai+] J. F. Hair, R. L. Tatham, R. E. Anderson, and W. Black. *Multivariate Data Analysis (5th Edition)*. 5th. Prentice Hall.

[Ham94] J. D. Hamilton. *Time series analysis*. Princenton University Press, 1994.

[Han+17] K. Han, W. Wan, H. Yao, and L. Hou. "Image Crowd Counting Using Convolutional Neural Network and Markov Random Field". In: 2017.

[Har+14] M. Harandi, M. Salzmann, and F. Porikli. "Bregman divergences for infinite dimensional covariance matrices". In: *IEEE Computer Vision and Pattern Recognition - CVPR*. 2014.

[HD12] M. Hoai and F. De la Torre. "Max-Margin Early Event Detectors". In: *IEEE Computer Vision and Pattern Recognition - CVPR*. 2012.

[HG17a] Z. Huang and L. V. Gool. "A Riemannian Network for SPD Matrix Learning". In: *AAAI International Conference on Aritificial Intelligence - AAAI*. 2017.

[HG17b] Z. Huang and L. V. Gool. "A Riemannian Network for SPD Matrix Learning". In: *AAAI International Conference on Aritificial Intelligence - AAAI*. 2017.

[Hin+12] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. “Improving neural networks by preventing co-adaptation of feature detectors”. In: *arXiv:1207.0580*. 2012.

[HK14] D.-A. Huang and K. M. Kitani. “Action-Reaction: Forecasting the Dynamics of Human Interaction”. In: *European Conference on Computer Vision*. 2014.

[HL15] D. P. Helmbold and P. M. Long. “On the Inductive Bias of Dropout”. In: *Journal of Machine Learning Research - JMLR* - 16 (2015), pp. 3403–3454.

[HO00] A. Hyvärinen and E. Oja. “Independent Component Analysis: Algorithms and Applications”. In: *Neural Networks* 13.4 (5 2000), pp. 411 –430.

[HO14] C.-J. Hsieh and P. Olsen. “Nuclear norm minimization via active subspace selection”. In: *International Conference on Machine Learning - ICML*. 2014, pp. 575–583.

[Hot36] H. Hotelling. “Relations between two sets of variates”. In: *Biometrika* 28.3 (4 1936), pp. 321 –377.

[HR94] L. Hansen and C. Rasmussen. “Pruning from Adaptive Regularization”. In: *Neural Computation* 6.6 (1994), pp. 1222–1231.

[Hua+12] D. Huang, R. S. Cabral, and F. De la Torre. “Robust Regression”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 2012.

[Hua+14] G. Huang, S. Song, J. N. D. Gupta, and C. Wu. “Semi-supervised and unsupervised extreme learning machines”. In: *IEEE Transactions on Cybernetics* 44.12 (2014), pp. 2405–2417.

[Hua+17] Z. Huang, C. Wan, T. Probst, and L. V. Gool. “Deep Learning on Lie Groups for Skeleton-based Action Recognition”. In: *IEEE Computer Vision and Pattern Recognition - CVPR*. 2017.

[Hub64] P. J. Huber. “Robust estimation of a location parameter”. In: *Annals of Mathematics and Statistics* 35.1 (1964).

[Hus+13] M. Hussein, M. Torki, M. Gowayyed, and M. El-Saban. “Human Action Recognition Using a Temporal Hierarchy of Covariance Descriptors on 3D Joint Locations”. In: *International Joint Conference on Artificial Intelligence - IJCAI* (2013).

[HV15] B. D. Haeffele and R. Vidal. “Global Optimality in Tensor Factorization, Deep Learning, and Beyond”. In: *CoRR abs/1506.07540*. 2015.

[HV17] B. D. Haeffele and R. Vidal. “Structured Low-Rank Matrix Factorization: Global Optimality, Algorithms, and Applications”. In: *CoRR abs/1708.07850*. 2017.

[IS15] S. Ioffe and C. Szegedy. “Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift”. In: *International Conference on Machine Learning - ICML*. 2015.

[J́+10] H. Jégou, M. Douze, C. Schmid, and P. Pérez. “Aggregating local descriptors into a compact image representation”. In: *IEEE Computer Vision and Pattern Recognition - CVPR*. 2010.

[Jai+16] A. Jain, A. R. Zamir, S. Savarese, and A. Saxena. “Structural-RNN: Deep Learning on Spatio-Temporal Graphs”. In: *IEEE Computer Vision and Pattern Recognition - CVPR*. 2016.

[Jay+13] S. Jayasumana, R. Hartley, M. Salzmann, H. Li, and M. Harandi. "Kernel Methods on the Riemannian Manifold of Symmetric Positive Definite Matrices". In: *IEEE Computer Vision and Pattern Recognition - CVPR*. 2013.

[JF16] B. Jimmy and B. Frey. "Adaptive dropout for training deep neural networks". In: *Advances in Neural Information Processing Systems - NIPS*. 2016.

[Joh73] G. Johansson. "Visual perception of biological motion and a model for its analysis". In: *Perception and Psychophysics* 14 (1973), 201–211.

[JW17] J. Y. Junwu Weng Chaoqun Weng. "Spatio-Temporal Naive-Bayes Nearest-Neighbor for Skeleton-Based Action Recognition". In: *IEEE Computer Vision and Pattern Recognition - CVPR*. 2017.

[Kan+15] M. Kan, S. Shan, and X. Chen. "Bi-Shifting Auto-Encoder for Unsupervised Domain Adaptation". In: *IEEE International Conference on Computer Vision - ICCV*. 2015.

[KB14] D. Kingma and J. Ba. "Adam: A Method for Stochastic Optimization". In: *CoRR:1412.6980*. 2014.

[KC09] T. K. Kim and R. Cipolla. "Multiple Classifier Boosting for Perceptual Co-clustering of Images and Visual Features". In: *Advances on Neural Information and Processing Systems - NIPS*. 2009, pp. 841–848.

[Ke+17] Q. Ke, M. Bennamoun, S. An, F. Sohel, and F. Boussaid. "A New Representation of Skeleton Sequences for 3D Action Recognition". In: *IEEE Computer Vision and Pattern Recognition - CVPR*. 2017.

[Kes+16] A. Kessy, A. Lewin, and K. Strimmer. "Optimal Whitening and Decorrelation". In: *arXiv:1512.00809v4*. 2016.

[KF16] Y. Kong and Y. Fu. "Max-Margin Action Prediction Machine". In: *TPAMI* 38.9 (2016), pp. 1844–1858.

[KH09] A. Krizhevsky and G. Hinton. "Learning multiple layers of features from tiny images". In: *Master's thesis, Department of Computer Science, University of Toronto* (2009).

[Kha+13] I. Khan, P. Roth, A. Bais, and H. Bischof. "Semi-supervised image classification with huberized Laplacian Support Vector Machines". In: *International Conference on Engineering & Technology - ICET*. 2013.

[Kil11] J. M. Kilner. "More than one pathway to action understanding". In: *Trends in Cognitive Sciences* 15 (8 2011), pp. 352–357.

[Kit+12] K. M. Kitani, B. D. Ziebart, J. A. Bagnell, and M. Hebert. "Activity Forecasting". In: *European Conference on Computer Vision*. 2012.

[KK12] P. Kar and H. Karnick. "Random Feature Maps for Dot Product Kernels". In: 2012.

[KM03] A. Kriegl and P. W. Michor. "Differentiable perturbation of unbounded operators". In: *Mathematische Annalen* 327.1 (2003), pp. 191–201.

[Kon+06] D. Kong, D. Gray, and H. Tao. "A Viewpoint Invariant Approach for Crowd Counting." In: *IAPR International Conference on Pattern Recognition - ICPR*. 2006.

[Kon+16] P. Koniusz, A. Cherian, and F. Porikli. "Tensor Representation via Kernel Linearization for Action Recognition from 3D Skeletons". In: *ECCV*. 2016.

[Kri+12a] A. Krizhevsky, I. Sutskever, and G. E. Hinton. "ImageNet Classification with Deep Convolutional Neural Networks". In: *Advances on Neural Information and Processing Systems - NIPS*. 2012.

[Kri+12b] A. Krizhevsky, I. Sutskever, and G. E. Hinton. "ImageNet Classification with Deep Convolutional Neural Networks". In: *Advances on Neural Information and Processing Systems - NIPS*. 2012.

[KS13] H. Koppula and A. Saxena. "Anticipating Human Activities using Object Affordances for Reactive Robotic Response". In: *Robot Science and Systems*. 2013.

[Kue+11] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre. "HMDB: A Large Video Database for Human Motion Recognition". In: *IEEE International Conference on Computer Vision - ICCV*. 2011.

[Kul+06] B. Kulis, M. Sustik, and I. Dhillon. "Learning low-rank kernel matrices". In: *International Conference on Machine Learning - ICML*. Morgan Kaufmann, 2006, pp. 505–512.

[Kum+10] A. Kumar, P. Rai, and A. Daumè III. "Co-regularized spectral clustering with multiple kernels". In: *Advances on Neural Information and Processing Systems - NIPS- workshops*. 2010.

[Kum+11] A. Kumar, P. Rai, and A. Daumè III. "Co-regularized multi-view spectral clustering". In: *International Conference on Machine Learning - ICML*. 2011.

[KW07] R. Khardon and G. Wachman. "Noise Tolerant Variants of the Perceptron Algorithm". In: 8 (2007), pp. 227–248.

[Lan+14] T. Lan, T.-C. Chen, and S. Savarese. "A Hierarchical Representation for Future Action Prediction". In: *European Conference on Computer Vision*. 2014.

[LBY16] Z. G. Li and T. Boqing Yang. "Improved Dropout for Shallow and Deep Learning". In: *Advances in Neural Information Processing Systems - NIPS*. 2016.

[Le+13] Q. Le, T. Sarlos, and A. Smola. "Fastfood - Approximating Kernel Expansion in Loglinear Time". In: *International Conference on Machine Learning - ICML*. 2013.

[Lea] .

[LeC+89] Y. LeCun, B. Boser, J. Denker, D. Henderson, R. Howard, W. Hubbard, and L. Jackel. "Backpropagation applied to handwritten zip code recognition". In: *Neural Computation*. 1989, pp. 541–551.

[Lee+15] M. C. Lee, W. L. Chiang, and C. J. Lin. "Fast Matrix-Vector Multiplications for Large-Scale Logistic Regression on Shared-Memory Systems". In: *International Conference on Data Mining - ICDM*. 2015.

[Lee+17] I. Lee, D. Kim, S. Kang, and S. Lee. "Ensemble Deep Learning for Skeleton-based Action Recognition using Temporal Sliding LSTM Networks". In: *IEEE International Conference on Computer Vision - ICCV*. 2017.

[Lee13] D.-H. Lee. "Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks". In: *International Conference on Machine Learning - ICML- workshops*. 2013.

[Lei+05] B. Leibe, E. Seemann, and B. Schiele. "Pedestrian Detection in Crowded Scenes". In: *IEEE Computer Vision and Pattern Recognition - CVPR*. 2005.

[LF14] K. Li and Y. Fu. "Prediction of human activity by discovering temporal sequence patterns". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 36.8 (2014), pp. 1644–1657.

[Li+08] M. Li, Z. Zhang, K. Huang, and T. Tan. "Estimating the number of people in crowded scenes by MID based foreground segmentation and head-shoulder detection." In: *IAPR International Conference on Pattern Recognition - ICPR*. 2008.

[Li+10a] W. Li, Z. Zhang, and Z. Liu. "Action recognition based on a bag of 3d points". In: *IEEE Computer Vision and Pattern Recognition - CVPR- workshops*. 2010.

[Li+10b] W. Li, Z. Zhang, and Z. Liu. "Action recognition based on a bag of 3D points". In: *IEEE Computer Vision and Pattern Recognition - CVPR- workshops*. 2010.

[Li+12] K. Li, J. Hu, and Y. Fu. "Modeling complex temporal composition of actionlets for activity prediction". In: *European Conference on Computer Vision*. 2012.

[Li+17a] C. Li, Y. Hou, P. Wang, and W. Li. "Joint Distance Maps Based Action Recognition with Convolutional Neural Network". In: *IEEE Signal Processing Letters* 24.5 (2017), pp. 624–628.

[Li+17b] W. Li, L. Wen, M.-C. Chang, S. N. Lim, and S. Lyu. "Adaptive RNN Tree for Large-Scale Human Action Recognition". In: *IEEE International Conference on Computer Vision - ICCV*. 2017.

[Liu+16] J. Liu, A. Shahroudy, D. Xu, and G. Wang. "Spatio-Temporal LSTM with Trust Gates for 3D Human Action Recognition". In: *European Conference on Computer Vision*. 2016.

[Liu+17a] J. Liu, G. Wang, P. Hu, L.-Y. Duan, and A. C. Kot. "Global context-aware attention LSTM networks for 3D action recognition". In: *IEEE Computer Vision and Pattern Recognition - CVPR* (2017).

[Liu+17b] J. Liu, G. Wang, P. Hu, L.-Y. Duan, and A. C. Kot. "Global Context-Aware Attention LSTM Networks for 3D Action Recognition". In: *IEEE Computer Vision and Pattern Recognition - CVPR*. 2017.

[LLZ11] S. Lambert-Lacroix and L. Zwald. "Robust regression through the Huber's criterion and adaptive lasso penalty". In: *Electronic Journal of Statistics* 5 (2011), pp. 1015–1053.

[LN06] F. Lv and R. Nevatia. "Recognition and segmentation of 3D human actions using HMM and multi-class adaboost". In: *European Conference on Computer Vision*. 2006.

[Low04] D. G. Lowe. "Distinctive Image Features from Scale-Invariant Keypoints". In: *International Journal of Computer Vision - IJCV* - 60.2 (2004), pp. 91–110.

[Loy+13a] C. C. Loy, S. Gong, and T. Xiang. "From semi-supervised to transfer counting of crowds". In: *IEEE International Conference on Computer Vision - ICCV*. 2013.

[Loy+13b] C. C. Loy, S. Gong, and T. Xiang. "From Semi-Supervised to Transfer Counting of Crowds". In: *IEEE International Conference on Computer Vision - ICCV*. 2013.

[Loy+13c] C. Loy, K. Chen, S. Gong, and T. Xiang. "Crowd Counting and Profiling: Methodology and Evaluation". In: *Modeling, Simulation*

and Visual Analysis of Crowds. Vol. 11. The International Series in Video Computing. 2013, pp. 347–382.

[LP+15] D. Lopez-Paz, L. Bottou, B. Schölkopf, and V. Vapnik. "Unifying distillation and privileged information". In: 2015.

[Lu+11] H. Lu, K. N. Plataniotis, and A. N. Venetsanopoulos. "A survey of multilinear subspace learning for tensor data". In: *Pattern Recognition* 44.7 (2011), pp. 1540–1551.

[LV07] J. A. Lee and M. Verleysen. *Nonlinear Dimensionality Reduction*. 1st. Springer Publishing Company, Incorporated, 2007.

[LW02] A. Liaw and M. Wiener. "Classification and Regression by randomForest". In: *R Journal* 2.3 (2002), pp. 18–22.

[M+̈07] M. Müller, T. Röder, M. Clausen, B. Eberhardt, B. Krüger, and A. Weber. "HDM-05 doc." In: *Tech. Rep.* 2007.

[Ma+04] R. Ma, L. Li, W. Huang, and Q. Tian. "On pixel count based crowd density estimation for visual surveillance". In: *IEEE Computational Intelligence Society - CIS*. 2004.

[Ma+16] S. Ma, L. Sigal, and S. Sclaroff. "Learning Activity Progression in LSTMs for Activity Detection and Early Detection". In: *IEEE Computer Vision and Pattern Recognition - CVPR*. 2016.

[Mar+97] A. Marana, S. Velastin, L. Costa, and R. Lotufo. "Estimation of crowd density using image processing". In: *Image Processing for Security Applications*. 1997.

[MH08] L. Van der Maaten and G. Hinton. "Visualizing data using t-SNE". In: 9.2579-2605 (2008), p. 85.

[Min+11] Mingtao, Y. Jia, and S.-C. Zhu. "Parsing video events with goal inference and intent prediction". In: *IEEE International Conference on Computer Vision - ICCV*. 2011.

[Min+13] H. Q. Minh, L. Bazzani, and V. Murino. "A unifying framework for vector-valued manifold regularization and multi-view learning." In: *International Conference of Machine Learning - ICML*. 2013.

[Min+14a] H. Q. Minh, L. Bazzani, and V. Murino. "A Unifying Framework in Vector-valued Reproducing Kernel Hilbert Spaces for Manifold Regularization and Co-Regularized Multi-view Learning". In: *arXiv:1401.8066* (2014).

[Min+14b] H. Q. Minh, M. San Biagio, and V. Murino. "Log-Hilbert-Schmidt metric between positive definite operators on Hilbert spaces". In: *Advances in Neural Information Processing Systems - NIPS*. 2014.

[Min+16a] H. Q. Minh, L. Bazzani, and V. Murino. "A unifying framework in vector-valued reproducing kernel Hilbert spaces for manifold regularization and co-regularized multi-view learning". In: *Journal of Machine Learning Research - JMLR* - 17.25 (2016), pp. 1–72.

[Min+16b] H. Q. Minh, M. San Biagio, L. Bazzani, and V. Murino. "Approximate Log-Hilbert-Schmidt Distances Between Covariance Operators for Image Classification". In: *IEEE Computer Vision and Pattern Recognition - CVPR*. 2016.

[MM00] O. L. Mangasarian and D. R. Musicant. "Robust linear and support vector regression". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 22.9 (2000), pp. 950–955.

[MM17] P. Morerio and V. Murino. "Correlation Alignment by Riemannian Metric for Domain Adaptation". In: *CoRR:1705.08180*. 2017.

[Moe+06] T. B. Moeslund, A. Hilton, and V. Krüger. "A Survey of Advances in Vision-based Human Motion Capture and Analysis". In: *Elsevier Computer Vision and Image Undersanding - CVIU* 104.2 (2006), pp. 90–126.

[Mor+17] P. Morerio, J. Cavazza, R. Volpi, R. Vidal, and V. Murino. "Curriculum Dropout". In: *IEEE International Conference on Computer Vision - ICCV*. 2017.

[Mor+18] P. Morerio, J. Cavazza, and V. Murino. "Minimal-Entropy Correlation Alignment for Unsupervised Deep Domain Adaptation". In: *submitted to ICLR*. 2018.

[Mou+15] H. Mousavi, S. Mohammadi, A. Perina, R. Chellali, and V. Murino. "Analyzing Tracklets for the Detection of Abnormal Crowd Behavior". In: *IEEE Winter Conference on Applications of Computer Vision - WACV*. 2015.

[Mus+02a] I. Muslea, S. Minton, and C. Knoblock. "Active plus semi-supervised learning equals robust multiview learning". In: *International Conference on Machine Learning - ICML*. 2002.

[Mus+02b] I. Muslea, S. Minton, and C. Knoblock. "Adaptive view validation: A first step towards automatic view detection". In: *International Conference on Machine Learning - ICML*. 2002.

[Mus+06] I. Muslea, S. Minton, and C. Knoblock. "Active learning with multiple views". In: 27.1 (2006), 203–233.

[Mül07] M. Müller. "Dynamic Time Warping". In: *Information Retrieval for Music and Motion*. Ed. by S. B. Heidelberg. 2007.

[Nat+13] N. Natarajan, I. S. Dhillon, P. K. Ravikumar, and A. Tewari. "Learning with Noisy Labels". In: *Advances on Neural Information and Processing Systems - NIPS*. 2013.

[Net+11] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, and A. Y. Ng. "Reading Digits in Natural Images with Unsupervised Feature Learning". In: *Advances on Neural Information and Processing Systems - NIPS- workshops*. 2011.

[Ng+15] J. Y.-H. Ng, M. Hausknecht, S. Vijayanarasimhan, O. Vinyals, R. Monga, and G. Toderici. "Beyond Short Snippets: Deep Networks for Video Classification". In: *IEEE Computer Vision and Pattern Recognition - CVPR*. 2015.

[NG00] K. Nigam and R. Ghani. "Analyzing the effectiveness and applicability of co-training". In: *ACM Conference on Information and Knowledge Management*. 2000.

[OL13] O. Oreifej and Z. Liu. "HON4D: Histogram of Oriented 4D Normals for Activity Recognition from Depth Sequences". In: *IEEE Computer Vision and Pattern Recognition - CVPR*. 2013.

[Ola] C. Olah. *Calculus on Computational Graphs: Backpropagation*. http://colah.github.io/posts/2015-08-Backprop/.

[ORLS16] D. Oñoro-Rubio and R. J. López-Sastre. "Towards Perspective-Free Object Counting with Deep Learning". In: *ECCV*. 2016.

[Ozt+05] E. Oztop, D. Wolpert, and M. Kawato. "Mental State Inference using Visual Control Parameter". In: *Cognitive Brain Research* 22 (2005), pp. 129–151.

[Pan+08] Y. Pang, Y. Yuan, and X. Li. "Gabor-based region covariance matrices for face recognition". In: 18.7 (2008), pp. 989–993.

[PCSC14] L. Lo Presti, M. La Cascia, S. Sclaroff, and O. Camps. "Gesture Modeling by Hanklet-based Hidden Markov Model". In: *Asian Conference on Computer Vision*. 2014.

[PD07] F. Perronnin and C. Dance. "Fishers on visual vocabularies for image categorization". In: *IEEE Computer Vision and Pattern Recognition - CVPR*. 2007.

[Pea01] K. Pearson. "LIII. On lines and planes of closest fit to systems of points in space". In: *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science* 2.11 (1901), pp. 559–572.

[Pen+06] X. Pennec, P. Fillard, and N. Ayache. "A Riemannian Framework for Tensor Computing". In: 66.1 (2006), pp. 41–66.

[RB06] V. Rabaud and S. Belongie. "Counting Crowded Moving Objects". In: *IEEE Computer Vision and Pattern Recognition - CVPR*. 2006.

[Rec+10] B. Recht, M. Fazel, and P. A. Parrilo. "Guaranteed Minimum-Rank Solutions of Linear Matrix Equations via Nuclear Norm Minimization". In: *Society for Industrial and Applied Mathematics, Review of - SIREV* 52.3 (2010), pp. 471–501.

[Ren+14] S. J. Rennie, V. Goel, and S. Thomas. "Annealed dropout training of deep networks". In: *Proceedings onf the IEEE Workshop on Speech, Language and Translation*. 2014, pp. 159–164.

[Ric88] J. Rice. *Mathematical statistics and data analysis*. Wadsworth & Brooks/Cole statistics/probability series. Brooks/Cole Pub. Co., 1988.

[Rif+11] S. Rifai, X. Glorot, B. Yoshua, and P. Vincent. "Adding noise to the input of a model trained with a regularized objective". In: *arXiv:1104.3250v1*. 2011.

[Rin] "An approximation of the Gaussian RBF kernel for efficient classification with SVMs". In: *Pattern Recognition Letters* 84 (2016), pp. 107 –113.

[RL05] P. Rousseeuw and A. Leroy. *Robust Regression and Outlier Detection*. Wiley Series in Probability and Statistics. Wiley, 2005.

[Roh+12] M. Rohrbach, S. Amin, M. Andriluka, and B. Schiele. "A Database for Fine Grained Activity Detection of Cooking Activities". In: *IEEE Computer Vision and Pattern Recognition - CVPR*. 2012.

[Roz+16] A. Rozantsev, M. Salzmann, and P. Fua. *Beyond Sharing Weights for Deep Domain Adaptation*. 2016.

[RR07] A. Rahimi and B. Recth. "Random Features for Large-Scale Kernel Machines". In: *Advances on Neural Information and Processing Systems - NIPS*. 2007.

[Rud66] W. Rudin. *Real and Complex Analysis*. Ed. by McGraw-Hill. 1966.

[Rya+15] D. Ryan, S. Denman, S. Sridharan, and C. Fookes. "An evaluation of crowd counting methods, features and regression models". In: *Elsevier Computer Vision and Image Undersanding - CVIU* 130 (2015), pp. 1 –17.

[Ryo+15] M. S. Ryoo, T. J. Fuchs, L. Xia, J. K. Aggarwal, and L. Matthies. "Robot-Centric Activity Prediction from First-Person Videos: What Will They Do to Me?" In: *Human Robot Interaction*. 2015.

[Ryo11] M. S. Ryoo. "Human activity prediction: Early recognition of ongoing activities from streaming videos". In: *IEEE International Conference on Computer Vision - ICCV*. 2011.

[Sae+10] K. Saenko, B. Kulis, M. Fritz, and T. Darrell. "Adapting Visual Category Models to New Domains". In: *European Conference on Computer Vision*. 2010.

[Sai+17] K. Saito, Y. Ushiku, and T. Harada. "Asymmetric Tri-training for Unsupervised Domain Adaptation". In: *International Conference on Machine Learning - ICML*. 2017.

[Sam+17] D. B. Sam, S. Surya, and R. V. Babu. "Switching convolutional neural network for crowd counting". In: *IEEE Computer Vision and Pattern Recognition - CVPR* (2017).

[San+13] A. Sanin, C. Sanderson, M. T. Harandi, and B. C. Lovell. "Spatio-temporal covariance descriptors for action and gesture recognition". In: *IEEE Winter Conference on Applications of Computer Vision - WACV*. 2013.

[Sau+98] C. Saunders, A. Gammerman, and V. Vovk. "Ridge Regression Learning Algorithm in Dual Variables". In: *International Conference on Machine Learning - ICML*. 1998.

[SB+13] M. San Biagio, M. Crocco, M. Cristani, S. Martelli, and V. Murino. "Heterogeneous Auto-similarities of Characteristics (HASC): Exploiting Relational Information for Classification". In: *IEEE International Conference on Computer Vision - ICCV*. 2013.

[Sch15] J. Schmidhuber. "Deep Learning in Neural Networks: An Overview". In: *Neural Networks* 61 (2015), pp. 85–117.

[Sei+13] L. Seidenari, V. Varano, S. Berretti, A. D. Bimbo, and P. Pala. "Recognizing Actions from Depth Cameras as Weakly Aligned Multi-part Bag-of-Poses". In: *IEEE Computer Vision and Pattern Recognition - CVPR- workshops*. 2013.

[SG08] K. Schindler and L. J. V. Gool. "Action snippets: How many frames does human action recognition require?" In: *IEEE Computer Vision and Pattern Recognition - CVPR*. 2008.

[Sha+16] A. Shahroudy, J. Liu, T.-T. Ng, and G. Wang. "NTU RGB+D: A Large Scale Dataset for 3D Human Activity Analysis". In: *IEEE Computer Vision and Pattern Recognition - CVPR*. 2016.

[She+05] Y. Sheikh, M. Sheikh, and M. Shah. "Exploring the Space of a Human Action". In: *IEEE International Conference on Computer Vision - ICCV*. 2005.

[She+13] S. Shekhar, V. M. Patel, H. V. Nguyen, and R. Chellappa. "Generalized Domain-Adaptive Dictionaries". In: *IEEE Computer Vision and Pattern Recognition - CVPR*. 2013.

[Sil+12] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus. "Indoor segmentation and support infer- ence from rgbd images". In: *European Conference on Computer Vision*. 2012.

[Sin+05] V. Sindhwani, P. Niyogi, and M. Belkin. "A co-regularization approach to semi-supervised learning with multiple views". In: *International Conference on Machine Learning - ICML- workshops*. 2005.

[SL12] G. Seber and A. Lee. *Linear Regression Analysis*. Wiley Series in Probability and Statistics. Wiley, 2012.

[SM09] D. Salomon and G. Motta. *Handbook of Data Compression*. 5th. Springer Publishing Company, Incorporated, 2009.

[SM13] S. Stein and S. J. McKenna. "Combining embedded accelerometers with computer vision for recognizing food preparation activities". In: *ACM UbiComp*. 2013.

[Sol+13] M. Soltanolkotabi, E. Elhamifar, and E. J. Candès. "Robust Subspace Clustering". In: *CoRR:1301.2603*. 2013.

[Soo+12] K. Soomro, A. R. Zamir, and M. Shah. "UCF101: A Dataset of 101 Human Action Classes From Videos in The Wild". In: *CRCV-TR-12-01*. 2012.

[Soo+16] K. Soomro, H. Idrees, and S. Mubarak. "Predicting the Where and What of actors and actions through Online Action Localization". In: *IEEE Computer Vision and Pattern Recognition - CVPR*. 2016.

[Sor+15] B. Soran, A. Farhadi, and L. Shapiro. "Generating Notifications for Missing Actions: Don't Forget to Turn the Lights Off!" In: *IEEE International Conference on Computer Vision - ICCV*. 2015.

[SR09] G. Stempfel and L. Ralaivola. "Learning SVMs from Sloppily Labeled Data". In: *International Conference on Artificial Neural Networks - ICANN*. 2009.

[Sri+14] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. "Dropout: A Simple Way to Prevent Neural Networks from Overfitting". In: *Journal of Machine Learning Research - JMLR* - 15.1 (2014), pp. 1929–1958.

[Sri+15] N. Srivastava, E. Mansimov, and R. Salakhutdinov. "Unsupervised Learning of Video Representations using LSTMs". In: *CoRR:1502.04681*. 2015.

[SS02] B. Scholkopf and A. Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. Adaptive Computation and Machine Learning. 2002.

[SS16] B. Sun and K. Saenko. "Deep CORAL: Correlation Alignment for Deep Domain Adaptation". In: *European Conference on Computer Vision- workshops*. 2016.

[Sta+12] J. C. Stapel, S. Hunnius, and H. Bekkering. "Online prediction of others' actions: the contribution of target object, action context, and movement kinematics". In: *Psychological Research*. 2012.

[Ste+16] R. Stewart, M. Andriluka, and A. Y. Ng. "End-to-end people detection in crowded scenes". In: *IEEE Computer Vision and Pattern Recognition - CVPR*. 2016.

[Sun+16] B. Sun, J. Feng, and K. Saenko. "Return of Frustratingly Easy Domain Adaptation". In: *AAAI International Conference on Aritical Intelligence - AAAI*. 2016.

[Sun13] S. Sun. "A survey of multi-view machine learning". In: *Neural Computing and Applications* 23.7 (2013), pp. 2031–2038.

[Tai+17] Y. Taigman, A. Polyak, and L. Wolf. "Unsupervised Cross-Domain Image Generation". In: *International Conference on Learning Representations*. 2017.

[Tan+11a] B. Tan, J. Zhang, and L. Wang. "Semi-supervised Elastic net for pedestrian counting". In: *Pattern Recognition* 44 (2011), pp. 2297 –2304.

[Tan+11b] B. Tan, J. Zhang, and L. Wang. "Semi-supervised Elastic net for pedestrian counting". In: *Pattern Recognition* 44 (2011), pp. 2297–2304.

[Tan+15] H. Tan, Z. Ma, S. Zhang, Z. Zhan, B. Zhang, and C. Zhang. "Grassmann manifold for nearest points image set classification." In: *Pattern Recognition Letters* 68 (2015), pp. 190–196.

[Tan13] Y. Tang. "Deep Learning using Support Vector Machines". In: *ICML workshop*. 2013.

[TE11] A. Torralba and A. A. Efros. "Unbiased look at dataset bias". In: *CVPR*. 2011.

[Tf] *Tensorflow*. http:www.tensorflow.org.

[Tos+13] D. Tosato, M. Spera, M. Cristani, and V. Murino. "Characterizing Humans on Riemannian Manifolds". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35.8 (2013), pp. 1972–1984.

[Tra+15] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri. "Learning Spatiotemporal Features with 3D Convolutional Networks". In: *IEEE International Conference on Computer Vision - ICCV*. 2015.

[Tri+15] I. Triguero, S. Garcia, and F. Herrera. "SEG-SSC: A framework based on synthetic examples generation for self-labeled semi-supervised classification". In: *IEEE Transactions on Cybernetics* 45.4 (2015), pp. 622–634.

[Tuz+06a] O. Tuzel, F. Porikli, and P. Meer. "Region Covariance: A Fast Descriptor for Detection and Classification". In: *IEEE European Conference on Computer Vision - ECCV*. 2006.

[Tuz+06b] O. Tuzel, F. Porikli, and P. Meer. "Region covariance: A fast descriptor for detection and classification". In: *European Conference on Computer Vision*. 2006.

[Tuz+08] O. Tuzel, F. Porikli, and P. Meer. "Pedestrian Detection via Classification on Riemannian Manifolds". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 30.10 (2008), pp. 1713–1727.

[Tze+14] E. Tzeng, J. Hoffman, N. Zhang, K. Saenko, and T. Darrell. "Deep Domain Confusion: Maximizing for Domain Invariance". In: 2014.

[Tze+15a] E. Tzeng, J. Hoffman, T. Darrell, and K. Saenko. "Simultaneous deep transfer across domains and tasks". In: *IEEE International Conference on Computer Vision - ICCV*. 2015.

[Tze+15b] E. Tzeng, J. Hoffman, T. Darrell, and K. Saenko. "Simultaneous Deep Transfer Across Domains and Tasks". In: *IEEE International Conference on Computer Vision - ICCV*. 2015.

[Tze+17] E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell. "Adversarial Discriminative Domain Adaptation". In: *IEEE Computer Vision and Pattern Recognition - CVPR*. 2017.

[VC16] R. Vemulapalli and R. Chellapa. "Rolling Rotations for Recognizing Human Actions From 3D Skeletal Data". In: *IEEE Computer Vision and Pattern Recognition - CVPR*. 2016.

[Vem+10] S. Vempati, A. Vedaldi, A. Zisserman, and C. V. Jawahar. "Generalized RBF feature maps for Efficient Detection". In: *British Machine Vision Conference - BMVC*. 2010.

[Vem+14] R. Vemulapalli, F. Arrate, and R. Chellappa. "Human Action Recognition by Representing 3D Skeletons as Points in a Lie Group". In: *IEEE Computer Vision and Pattern Recognition - CVPR*. 2014.

[Vid+16a] R. Vidal, Y. Ma, and S. S. Sastry. *Generalized Principal Component Analysis*. 1st. Springer Publishing Company, Incorporated, 2016.

[Vid+16b] R. Vidal, Y. Ma, and S. S. Sastry. *Generalized Principal Component Analysis*. 1st. Springer Publishing Company, Incorporated, 2016. ISBN: 0387878106, 9780387878102.

[Von+16] C. Vondrick, H. Pirsiavash, and A. Torralba. "Anticipating Visual Representations with Unlabeled Video". In: *IEEE Computer Vision and Pattern Recognition - CVPR*. 2016.

[VW97] P. Viola and W. M. Wells III. "Alignment by Maximization of Mutual Information". In: *International Journal of Computer Vision - IJCV* - 24.2 (1997).

[VZ12] A. Vedaldi and A. Zisserman. "Efficient Additive Kernels via Explicit Feature Maps". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 34.3 (2012).

[Wag+13] S. Wager, S. Wang, and P. S. Liang. "Dropout Training as Adaptive Regularization". In: *Advances in Neural Information Processing Systems - NIPS*. 2013.

[Wag+14] S. Wager, W. Fithian, S. Wang, and P. S. Liang. "Altitude Training: Strong Bounds for Single-Layer Dropout". In: *Advances in Neural Information Processing Systems - NIPS*. 2014.

[Wal+14] J. Walker, A. Gupta, and M. Hebert. "Patch to the Future: Unsupervised Visual Prediction". In: *IEEE Computer Vision and Pattern Recognition - CVPR*. 2014.

[Wan+11] Z. Wang, B. Fan, and F. Wu. "Local Intensity Order Pattern for feature description". In: *IEEE International Conference on Computer Vision - ICCV*. 2011.

[Wan+12a] G. Wang, F. Wang, T. Chen, D. Y. Yeung, and F. Lochovsky. "Solution path for manifold regularized semi-supervised classification". In: *IEEE Transactions on Systems, Man and Cybernetics* 44.12 (2012), pp. 308 –319.

[Wan+12b] J. Wang, Z. Liu, Y. Wu, and J. Yuan. "Mining actionlet ensemble for action recognition with depth cameras". In: *IEEE Computer Vision and Pattern Recognition - CVPR*. 2012.

[Wan+12c] J. Wang, Z. Liu, J. Chorowski, Z. Chen, and Y. Wu. "Robust 3d action recognition with random occupancy patterns". In: *European Conference on Computer Vision*. 2012.

[Wan+12d] J. Wang, Z. Liu, Y. Wu, and J. Yuan. "Mining Actionlet Ensemble for Action Recognition with Depth Cameras". In: *IEEE Computer Vision and Pattern Recognition - CVPR*. 2012.

[Wan+12e] R. Wang, H. Guo, L. S. Davis, and Q. Dai. In: *IEEE Computer Vision and Pattern Recognition - CVPR*. 2012.

[Wan+12f] R. Wang, H. Guo, L. S. Davis, and Q. Dai. "Covariance discriminative learning: A natural and efficient approach to image set classification." In: *IEEE Computer Vision and Pattern Recognition - CVPR*. 2012.

[Wan+13a] L. Wan, M. Zeiler, S. Zhang, Y. L. Cun, and R. Fergus. "Regularization of Neural Networks using DropConnect". In: *International Conference on Machine Learning*. 2013.

[Wan+13b] C. Wang, Y. Wang, and A. L. Yuille. “An Approach to Pose-Based Action Recognition.” In: *IEEE Computer Vision and Pattern Recognition - CVPR*. 2013.

[Wan+13c] H. Wang, A. Kläser, C. Schmid, and C.-L. Liu. “Dense trajectories and motion boundary descriptors for action recognition”. In: 103.1 (2013), pp. 60–79.

[Wan+15a] C. Wang, H. Zhang, L. Yang, S. Liu, and X. Cao. “Deep people counting in extremely dense crowds”. In: *ACM Conference on Multimedia - ACMM*. 2015.

[Wan+15b] L. Wang, J. Zhang, L. Zhou, C. Tang, and W. Li. “Beyond Covariance: Feature Representation With Nonlinear Kernel Matrices”. In: *IEEE International Conference on Computer Vision - ICCV*. 2015.

[Wan+16] P. Wang, Z. Li, Y. Hou, and W. Li. “Action Recognition Based on Joint Trajectory Maps Using Convolutional Neural Networks”. In: *ACM Conference on Multimedia - ACMM*. 2016.

[Wen+17] J. Weng, C. Weng, and J. Yuan. “Spatio-Temporal Naive-Bayes Nearest-Neighbor (ST-NBNN) for Skeleton-Based Action Recognition”. In: *IEEE Computer Vision and Pattern Recognition - CVPR*. 2017.

[WG15] H. Wu and X. Gu. “Towards dropout training for convolutional neural networks”. In: *Neural Networks* 71 (2015), pp. 1–10.

[Wie+13] M. Wiering, M. Schutten, A. Millea, A. Meijster, and L. R. B. Schomaker. “Deep Support Vector Machines for Regression Problems”. In: *International Workshop on Advances in Regularization, Optimization, Kernel Methods, and Support Vector Machines: theory and applications*. 2013.

[WM13] S. Wang and C. D. Manning. “Fast Dropout Training”. In: *International Conference on Machine Learning - ICML*. 2013.

[WS13] H. Wang and C. Schmid. “Action Recognition with Improved Trajectories”. In: *IEEE International Conference on Computer Vision - ICCV*. 2013.

[Wu+06] X. Wu, G. Liang, K. K. Lee, and Y. Xu. “Crowd Density Estimation Using Texture Analysis and Learning”. In: *IEEE International Conference on Robotics and Biomimetics - ROBIO*. 2006.

[WZ07] W. Wang and Z. Zhou. “Analyzing co-training style algorithms”. In: *European Conference on Machine Learning - ECML*. 2007.

[WZ10] W. Wang and Z.-H. Zhou. “A new analysis of co-training”. In: *International Conference on Machine Learning - ICML*. 2010.

[Xia+12] L. Xia, C.-C. Chen, and J. Aggarwal. “View invariant human action recognition using histograms of 3D joints”. In: *IEEE Computer Vision and Pattern Recognition - CVPR- workshops*. 2012.

[Xie+13] D. Xie, S. Todorovic, and S.-C. Zhu. “Inferring “Dark Matter” and “Dark Energy" from Videos”. In: *IEEE International Conference on Computer Vision - ICCV*. 2013.

[Xio+17] F. Xiong, X. Shi, and D.-Y. Yeung. “Spatiotemporal modeling for crowd counting in videos”. In: *arXiv:1707.07890*. 2017.

[Xu+13] C. Xu, D. Tao, and C. Xu. “A Survey on Multi-view Learning”. In: *arXiv:1304.5634*. 2013.

[Xu+15a] Z. Xu, L. Qing, and J. Miao. "Activity Auto-Completion: Predicting Human Activities From Partial Videos". In: *IEEE International Conference on Computer Vision - ICCV*. 2015.

[Xu+15b] Z. Xu, Y. Yang, and A. G. Hauptmann. "A Discriminative CNN Video Representation for Event Detection". In: *IEEE Computer Vision and Pattern Recognition - CVPR*. 2015.

[Yam+11] K. Yamaguchi, A. C. Berg, L. E. Ortiz, and T. L. Berg. "Who are you with and where are you going?" In: *IEEE Computer Vision and Pattern Recognition - CVPR*. 2011.

[YT12] X. Yang and Y. Tian. "Eigenjoints-based action recognition using naivebayes-nearest-neighbor". In: *IEEE Computer Vision and Pattern Recognition - CVPR- workshops*. 2012.

[YT14a] X. Yang and Y. Tian. "Super normal vector for activity recognition using depth sequences". In: *IEEE Computer Vision and Pattern Recognition - CVPR*. 2014.

[YT14b] X. Yang and Y. Tian. "Super Normal Vector for Activity Recognition Using Depth Sequences". In: *IEEE Computer Vision and Pattern Recognition - CVPR*. 2014.

[Yu+07] S. Yu, B. Krishnapuram, R. Rosales, and R. Rao. "Bayesian cotraining". In: *Advances on Neural Information and Processing Systems - NIPS*. 2007.

[Yu+11] S. Yu, B. Krishnapuram, R. Rosales, and R. Rao. "Bayesian cotraining". In: vol. 12. 2011, 2649–2680.

[Yu+12] G. Yu, J. Yuan, and Z. Liu. "Predicting Human Activities using Spatio-Temporal Structure of Interest Points". In: *ACM Conference on Multimedia - ACMM*. 2012.

[Yua+07] M. Yuan, A. Ekici, Z. Lu, and R. Monteiro. "Dimension reduction and coefficient estimation in multivariate linear regression". In: *Journal of the Royal Statistical Society* 69.3 (2007), pp. 329–346.

[YVG14] A. Yao and P. Van Gool L. Kohli. "Gesture recognition portfolios for personalization". In: *IEEE Computer Vision and Pattern Recognition - CVPR*. 2014.

[Zan+13] M. Zanfir, M. Leordeanu, and C. Sminchisescu. "The Moving Pose: An Efficient 3D Kinematics Descriptor for Low-Latency Action Recognition and Detection". In: *IEEE International Conference on Computer Vision - ICCV*. 2013.

[ZB15] M. Ziaeefard and R. Bergevin. "Semantic human activity recognition: A literature review". In: *Pattern Recognition* 48.8 (2015), pp. 2329 –2345.

[ZF14] M. D. Zeiler and R. Fergus. "Visualizing and Understanding Convolutional Networks". In: *European Conference on Computer Vision*. 2014.

[Zha+15a] S. Zha, F. Luisier, W. Andrews, N. Srivastava, and R. Salakhutdinov. "Exploiting Image-trained CNN Architectures for Unconstrained Video Classification". In: *British Machine Vision Conference - BMVC*. 2015.

[Zha+15b] C. Zhang, H. Li, X. Wang, and X. Yang. "Cross-scene crowd counting via deep convolutional neural networks". In: *IEEE Computer Vision and Pattern Recognition - CVPR*. 2015.

[Zha+16a] X. Zhang, Y. Wang, M. Gou, M. Sznaier, and O. Camps. "Efficient Temporal Sequence Comparison and Classification Using Gram

Matrix Embeddings on a Riemannian Manifold". In: *IEEE Computer Vision and Pattern Recognition - CVPR*. 2016.

[Zha+16b] Y. Zhang, D. Zhou, S. Chen, S. Gao, and Y. Ma. "Single-image crowd counting via multi-column convolutional neural network". In: *IEEE Computer Vision and Pattern Recognition - CVPR*. 2016.

[Zhi+16] H. Zhicheng, J. Jie, L. Caihua, W. Yuan, Y. Airu, and H. Yalou. "Dropout Non-negative Matrix Factorization for Independent Feature Learning". In: *Natural Language Understanding and Intelligent Applications - NLPCC*. 2016.

[Zho+14] B. Zhou, A. Lapedriza, J. Xiao, A. Torralba, and A. Oliva. "Learning Deep Features for Scene Recognition using Places Database". In: *Advances on Neural Information and Processing Systems - NIPS*. 2014.

[Zho+17] L. Zhou, L. Wang, J. Zhang, Y. Shi, and Y. Gao. "Revisiting Distance Metric Learning for SPD Matrix based Visual Representation". In: *IEEE Conference on Computer Vision and Pattern Recognition - CVPR*. 2017.

[Zun+17a] A. Zunino, J. Cavazza, A. Koul, A. Cavallo, C. Becchio, and V. Murino. "Predicting Human Intentions from Motion Cues Only: A 2D+3D Fusion Approach". In: *GIRPR International Conference on Image Analysis and Processing - ICIAP*. 2017.

[Zun+17b] A. Zunino, J. Cavazza, and V. Murino. "Revisiting Human Action Recognition". In: *GIRPR International Conference on Image Analysis and Processing - ICIAP*. 2017.

[Zun+17c] A. Zunino, J. Cavazza, A. Koul, A. Cavallo, C. Becchio, and V. Murino. "What Will I Do Next? The Intention from Motion Experiment". In: *IEEE Computer Vision and Pattern Recognition - CVPR- workshops*. 2017.

[ZZ15] S. Zhai and Z. M. Zhang. "Dropout Training of Matrix Factorization and Autoencoders for Link Prediction in Sparse Graphs". In: *arXiv:1512.04483v1*. 2015.